



ulm university universität
uulm

Universität Ulm | 89069 Ulm | Germany

**Fakultät für
Ingenieurwissenschaften,
Informatik und
Psychologie**
Institut für Datenbanken
und Informationssysteme

Konzept und Design einer grafischen Benutzeroberfläche eines datenzentrierten Prozess-Management-Systems

Bachelorarbeit an der Universität Ulm

Vorgelegt von:

Angelina Henschel
angelina.bangert@uni-ulm.de

Gutachter:

Prof. Dr. Manfred Reichert

Betreuer:

Kevin Andrews

2015

Fassung 10. März 2016

Zusammenfassung

Der zunehmende wirtschaftliche Druck im heutigen Informationszeitalter hängt vom Umgang mit betrieblichen Informationen ab. Informationen sind heutzutage ein wichtiger Faktor für den Unternehmenserfolg. Deshalb setzen immer mehr Unternehmen Business Prozess Management Systeme ein, die die Geschäftsprozesse unterstützen sollen. Erleichtert die verwendete Anwendung in einem Unternehmen nicht die Arbeitsabläufe, kann keine effiziente, effektive als auch zufriedenstellende Arbeit gewährleistet werden. Mit Hilfe eines roten Fadens der sich durch die komplette Benutzeroberfläche zieht, kann die Arbeit des Benutzers unterstützt werden. Im Rahmen dieser Arbeit wurde ein Usability-Konzept für die Laufzeitumgebung des PHILharmonic Flows Framework ausgearbeitet. Dies beinhaltet die Gestaltung einer geeigneten Navigationsstruktur, eine effiziente Aufteilung der Benutzeroberfläche und eine harmonische Farbwahl sowohl für die Benutzeroberfläche als auch den Mikro-Prozess Graphen. Mit definierten Usability Zielen unter Beachtung der Normen und Vorgehensweisen wurde die Benutzeroberfläche gestaltet.

VORBEMERKUNG ZUM SPRACHGEBRAUCH

Alle Personen- und Funktionsbezeichnungen in dieser Arbeit gelten für Frauen und Männer in gleicher Weise. Bei Bezeichnungen von Personen und Personengruppen wird auf die Darstellung der weiblichen Schreibweise verzichtet, um eine bessere Lesbarkeit zu gewähren. Mit der männlichen Form, wie z.B. *Benutzer*, sind stets auch alle weiblichen Personen angesprochen.

Inhalt

1 Einleitung.....	1
1.1 Motivation	1
1.2 Ziel der Arbeit	2
1.3 Aufbau der Arbeit.....	2
2 Grundlagen Usability	3
2.1 Begriffsdefinition	3
2.1.1 System	3
2.1.2 Nutzungskontext	3
2.1.3 Usability	3
2.1.4 Usability Experience (UX)	4
2.1.5 Usability Engineering	4
2.1.6 Zielgruppe	4
2.1.7 Benutzeroberfläche.....	4
2.1.8 Mockup	4
2.1.9 Prototyp.....	4
2.2 Normen und Designrichtlinien	4
2.2.1 DIN EN ISO 9241 : Ergonomie der Mensch-System-Interaktion.....	5
2.3 Styleguides	6
2.4 Psychologische Grundlagen der Systementwicklung	8
2.4.1 Visuelle Wahrnehmung.....	9
2.4.2 Wirkung von Farbe.....	10
3 Anforderungsanalyse.....	12
3.1 Anforderungsanalyse.....	13
3.2 Geschäfts- und Einsatzziele	13
3.3 Analyse Ist-Stand	13
3.4 Kontextanalyse	13
3.5 Benutzerprofilanalyse	13

3.6	Aufgabenanalyse	14
3.7	Umgebungsbedingungen	15
3.8	Hardware- Software- Randbedingungen	15
3.9	Generelle Entwurfsprinzipien	15
3.10	Usability-Ziele	15
3.11	Zusammenfassung	17
4	User-Interface-Entwurf	18
4.2	Workflow Reengineering	19
4.3	Konzeptuelles User-Interface-Modell	19
4.4	User-Interface Mock-Ups	19
4.5	Iterative User-Interface-Walkthroughs	20
4.6	Elektronische User-Interface-Prototypen	20
4.7	Iterative Usability-Tests	20
4.8	User-Interface Styleguide	20
4.9	Detailentwurf	20
4.10	Zusammenfassung	21
5	PHILharmonicFlows Konzept	22
5.1	Überblick über PHILharmonic Flows	23
5.2	Datenstruktur	24
5.2.1	Objekttypen	24
5.2.2	Relationen	24
5.3	Prozessstruktur	25
5.3.1	Mikro-Prozess	25
5.3.2	Zustände	27
5.3.3	Mikroschritt	27
5.3.4	Makro-Prozess	28
5.4	Zusammenfassung	29

6 Usability-Konzept.....	30
6.1 Allgemeiner Aufbau	30
6.1.1 Die Sidebar	32
6.1.2 Anmeldung.....	33
6.1.3 Startbildschirm	33
6.1.4 Ansicht Objektverwaltung.....	36
6.1.5 Ansicht Ausführung Administratorsicht	38
6.1.6 Ansicht Objektstruktur	40
6.1.7 Der Mikro-Prozess Graph	41
6.2 Farbkonzept	45
6.3 Schrift	49
6.4 Symbolik.....	50
7 Zusammenfassung und Ausblick.....	52
A Mikro-Prozess Graph	53
Abbildungsverzeichnis	55
Tabellenverzeichnis	57
Literaturverzeichnis	58

1

Einleitung

*„ Wer die Prozesse im Unternehmen nicht beherrscht,
beherrscht das ganze Unternehmen nicht“.*

William Edwards Deming (*1900, † 1993)

1.1 Motivation

Der zunehmende wirtschaftliche Druck im heutigen Informationszeitalter hängt vom Umgang mit betrieblichen Informationen ab. Informationen sind heutzutage ein wichtiger Faktor für den Unternehmenserfolg. Der Zugang zu Informationen wird immer leichter und preiswerter. Im Gegensatz dazu ist die Menge der Daten die zur Verfügung stehen so groß, dass der Aufwand der Auswertung zu viel Zeit in Anspruch nimmt. Trotz der zunehmenden Reife der softwaregestützten Informationssysteme werden Geschäftsprozesse immer noch nicht ausreichend unterstützt [1].

Herkömmliche Prozess-Management-Systeme, die zum größten Teil aktivitätsgetrieben sind, unterstützen die Modellierung, Überwachung und Ausführung von Geschäftsprozessen. Die Abfolgen der Aktivitäten im Prozess sind fest vorgegeben. Das macht die ausgeführten Prozesse unflexibel und ineffizient [1].

In bereits existierenden datenzentrierten Prozess-Management-Systemen wird versucht, eine flexiblere Struktur der Aktivitäten mit Hilfe der Integration von Daten, Prozessen und Benutzern zu schaffen. Beispielsweise ist der Benutzer nicht auf den Fortschritt des Prozesses angewiesen, d.h. Daten können schon zur Verfügung gestellt werden obwohl sie noch nicht gebraucht werden.

PHILharmonicFlows ist ein solches datenzentriertes Prozess-Management-System, welches an der Universität Ulm entwickelt worden ist.

1.2 Ziel der Arbeit

Im Rahmen dieser Arbeit soll ein Usability-Konzept für die Laufzeitumgebung des PHILharmonicFlows Framework ausgearbeitet werden. Dies beinhaltet die Gestaltung geeigneter Navigationselemente, eine effiziente Aufteilung der Benutzeroberfläche und eine harmonische Farbwahl sowohl für die Benutzeroberfläche als auch den Mikro-Prozess Graphen. Ein Mikro-Prozess steuert den Zustand eines Objekts und der Mikro-Prozess Graph überwacht diesen Prozess. Die Benutzeroberfläche soll den Benutzer bei seinen Aufgaben unterstützen und einfach bedienbar sein. Die Implementierung des Konzepts ist nicht Bestandteil dieser Bachelorarbeit. Die Benutzeroberfläche der Laufzeitumgebung erlaubt die Instanziierung und Ausführung von Mikro-Prozessen. Die Interaktion mit den datengetriebenen Mikro-Prozessen findet über automatisch generierte Formulare statt. Zusätzlich kann die Benutzeroberfläche das Fortschreiten einer Mikro-Prozessinstanz anhand eines Graphen veranschaulicht werden. Es muss bei der Gestaltung darauf geachtet werden, dass es verschiedenen Benutzerrechte gibt. Dies hat nämlich Auswirkungen auf die Anordnung der Steuer- und Navigationselemente und muss im Design berücksichtigt werden. Mit definierten Usability Zielen soll unter Beachtung der Normen und Vorgehensweisen (vgl. Kapitel 2.2) die Benutzeroberfläche gestaltet werden.

1.3 Aufbau der Arbeit

Kapitel 2 bildet die Grundlage dieser Arbeit, in der wichtige Begriffe der Usability erläutert werden. Die beschriebenen Normen und Designrichtlinien werden bei der Gestaltung der Benutzeroberfläche berücksichtigt. Abgeschlossen wird das zweite Kapitel mit den psychologischen Grundlagen der Systementwicklung, wie z.B. der visuellen Wahrnehmung und Bedeutung von Farben.

Im dritten Kapitel werden die wesentlichen Anforderungen zur Gestaltung des Gesamtsystems festgehalten. Mit Hilfe der Ergebnisse der Anforderungsanalyse werden im vierten Kapitel die Voraussetzungen für die Gestaltung der Benutzeroberfläche festgelegt.

Kapitel 5 bietet eine Übersicht über das PHILharmonicFlows Konzept, welche notwendig ist, um den Inhalt dieser Arbeit zu verstehen. Aufbauend auf den vorherigen Kapiteln wird in Kapitel 6 das visuelle Design der Benutzeroberfläche erläutert. Eine Zusammenfassung und ein Ausblick runden die Arbeit in Kapitel 7 ab.

2

Grundlagen Usability

„Usability is like cooking: everybody needs the results, anybody can do it reasonably well with a bit of training, and yet it takes a master to produce a gourmet outcome.“

Jakob Nielsen (*1957)

2.1 Begriffsdefinition

In diesem Kapitel werden einige zentrale Begriffe, die für das Thema Usability von zentraler Bedeutung sind, näher erläutert.

2.1.1 System

Ein System ist eine Sammlung von Komponenten die zum Erfüllen einer bestimmten Funktion oder Menge von Funktionen zusammengeschaltet werden [2]. Beispiele für solche Komponenten können Hardware, Software oder Benutzer sein.

2.1.2 Nutzungskontext

Der Nutzungskontext enthält den Benutzer, Arbeitsaufgaben, Arbeitsmittel (Hardware, Software, Materialien) und die physische und soziale Umgebung, in der das Produkt genutzt wird [3]. Der Nutzungskontext ist ein Kernbegriff bei der Definition der Gebrauchstauglichkeit. Definiert wird dieser in der DIN ISO 9241 Teil 11: Anforderung an die Gebrauchstauglichkeit.

2.1.3 Usability

Dabei bezeichnet *Effektivität* die Genauigkeit, ob und in welchem Ausmaß die Endbenutzer ihre Ziele erreichen können. Die *Effizienz* misst den Aufwand, der zur Erreichung dieses Ziels nötig ist. Der Parameter *Zufriedenheit*, womit die positive Erfahrung gemeint ist, die ein Benutzer beim Benutzen eines Systems macht.

Diese drei Leitkriterien werden in der DIN EN ISO 9241-11 festgelegt, orientieren sich am Benutzer und sind vom Nutzungskontext einer Software abhängig. Diese Eigenschaften stellen die Mindestanforderung dar, welche ein interaktives System erfüllen sollte.

2.1.4 Usability Experience (UX)

User Experience (dt. Benutzererlebnis bzw. Nutzungserfahrung) erweitert den Begriff Usability um „ästhetische und emotionale Faktoren die ein Nutzer rund um die Nutzung einer Anwendung machen kann. UX ist die Erfahrung, die ein Nutzer mit einem Produkt machen kann.

2.1.5 Usability Engineering

„Usability Engineering (dt. Software- Ergonomie) ist die Entwicklung von Nutzungsanforderungen, Prototypen und Softwareprodukten sowie deren Validierung und Verbesserung unter systematischen Anwendung von Usability-Methoden im Rahmen des benutzerzentrierten Entwicklungsprozesses“ [4]. Es wird die Usability eines Produkts festgelegt, gemessen und verbessert.

2.1.6 Zielgruppe

Es handelt sich um die Gruppe von Endbenutzern, nach dessen Wünschen, Vorstellungen und Bedürfnissen das System entwickelt wird.

2.1.7 Benutzeroberfläche

Eine Benutzeroberfläche ist eine Schnittstelle, über die ein Benutzer Software oder Hardware verwenden kann. Sie sollte intuitiv verwendbar und benutzerfreundlich gestaltet sein, gerade für Zielgruppen mit unerfahrenen Benutzern die nicht die notwendigen Fachkenntnisse besitzen.

2.1.8 Mockup

Ein Mockup (dt. Statische Designvorlage) ist eine grobe Skizze der zu entwickelnden Oberfläche. Es visualisiert den Inhalt und demonstriert die wesentliche Funktionalität in einer statischen Art, stellt die Struktur der Informationen dar und unterstützt die gestalterische Prüfung des Layouts. [5]

2.1.9 Prototyp

Ein Prototyp, zeigt die beschlossene Funktionalität und Benutzerfreundlichkeit der Oberfläche. Anhand dessen kann vorab die Funktionalität der zukünftigen Oberfläche oder einzelnen Elementen getestet werden. [5]

2.2 Normen und Designrichtlinien

Seit Anfang der 1980er Jahre werden Normen für die Gestaltung von Benutzerschnittstellen definiert. [6] Der Einsatz von Normen und Richtlinien gibt dem Entwickler

einen Orientierungsrahmen und stellt zugleich ein Hilfsmittel dar um benutzbare, gebrauchstaugliche Software zum Nutzen des Benutzers zu gestalten. [6] „Normen sind sehr stark bindend und haben in manchen Ländern gesetzlichen Charakter“ [7].

Bei den im folgenden Kapitel 2.2.1 vorgestellten Normen, handelt es sich um Normen der internationalen Standardisierungsorganisation (kurz: ISO). In Tabelle 1 wird die ISO 924 – 110, welche sich auf die Grundsätze der Dialoggestaltung beziehen, näher erläutert [6]. Styleguides (dt. Gestaltungsrichtlinien) haben einen nicht so stark bindenden Charakter wie Normen [7]. Grundlegend werden hier konkrete Aussagen über das Aussehen von Merkmalen der Benutzerschnittstelle festgehalten [8].

2.2.1 DIN EN ISO 9241 : Ergonomie der Mensch-System-Interaktion

Die DIN EN ISO 9241 beschreibt als Standard die Ergonomie der Mensch-System-Interaktion. Ziel der Richtlinie ist es dem Benutzer die Ausführung von Aufgaben zu erleichtern und gesundheitliche Schäden beim Arbeiten am Bildschirm zu vermeiden. Ein gut gestalteter Dialog sollte nach DIN EN ISO 9241- Teil 110 – *Grundsätze der Dialoggestaltung* die folgenden Eigenschaften erfüllen [6] :

Grundsätze	Beschreibung
<i>Aufgabenangemessenheit</i>	Der Benutzer soll bei der Erledigung seiner Arbeitsaufgabe unterstützt werden, seine Aufgaben effektiv und effizient zu erledigen
<i>Selbstbeschreibungsfähigkeit</i>	Jeder einzelne Dialogschritt ist durch Beschreibung oder Rückmeldung unmittelbar verständlich oder er wird auf Anfrage des Benutzers erklärt.
<i>Steuerbarkeit</i>	Der Benutzer kann die Geschwindigkeit und Richtung des Dialogablaufs starten und lenken, bis das gewünschte Ziel erreicht ist.
<i>Erwartungskonformität</i>	Die Erwartungskonformität beinhaltet konsistente Dialoge, welche den Benutzerkenntnissen entsprechen.
<i>Fehlertoleranz</i>	Bei einer Fehleingabe kann der Benutzer mittels minimalen Korrekturaufwandes den Fehler beheben.

<i>Individualisierbarkeit</i>	Das System ist individualisierbar, wenn es sich an die Vorlieben und Fähigkeiten des Benutzers anpassen lässt.
<i>Lernförderlichkeit</i>	Der Benutzer wird während des Lernprozesses bei der Bedienung des Systems angeleitet und unterstützt.

Tabelle 1: Grundsätze der Dialoggestaltung [6]

2.3 Styleguides

Das Ziel der Definition eines Styleguide ist es, eine konsistente Benutzungsoberfläche für komplexe Anwendungen schaffen zu können [6]. Bevor Normen und Gesetze veröffentlicht wurden, haben Experten der Softwareergonomie Richtlinien aufgestellt, die dabei helfen, die ergonomische Qualität von Benutzungsschnittstellen zu erhöhen. Ein Pionier auf diesem Gebiet ist Ben Shneiderman. Er hat acht goldene Regeln für eine gut gestaltete Benutzungsschnittstelle erstellt [6]. Ebenso der Softwareergonom Jakob Nielsen, der zehn Usability-Heuristiken aufstellte [6]. Shneiderman und Nielsens Richtlinien überschneiden sich teilweise in ihren Empfehlungen, als auch in den vorher aufgeführten Normen.

In Tabelle 2 [9] werden die zehn wichtigsten Usability-Heuristiken vorgestellt und betrachtet [9]. Da die Oberfläche von PHILharmonicFlows auf den Windows 10 Designprinzipien aufbaut, werden diese in Tabelle 2 vorgestellt.

Allgemeine Usability-Heuristiken

Richtlinien	Beschreibung
<i>Sichtbarkeit des Systemstatus</i>	Das System sollte den Benutzer stets über den aktuellen Systemstatus informieren, indem es Feedback in angemessener Zeit gibt.
<i>Übereinstimmung zwischen dem System und der realen Welt</i>	Das System sollte Fachbegriffe vermeiden, oder sind für den Laien verständlich zu erklären.
<i>Benutzerkontrolle und -freiheit</i>	Fehlbedienungen seitens des Benutzers darf das System nicht zulassen, insbesondere solche, die das System in einen Zustand bringen, aus dem der Benutzer nicht selbst wieder herausfinden kann.

<i>Konsistenz und Standards</i>	Dem Benutzer sollten ähnliche Dinge immer auf gleicher Art und Weise dargestellt werden, z.B. in dem das System in gleichen Situationen oder Aktionen dieselben Konventionen verwendet.
<i>Fehlervermeidung</i>	Das System sollte versuchen Fehler erst gar nicht auftreten zu lassen.
<i>Erkennen statt Erinnern</i>	Die Kapazität des Kurzzeitgedächtnisses eines Benutzers ist begrenzt. Der Benutzer sollte sich nicht an Informationen erinnern müssen, die in einem anderen Bereich wichtig waren.
<i>Flexibel und Effizienz</i>	Die Abläufe eines Systems sollten für Novizen als auch für Experten so gestalten sein, dass beide sie effizient bedienen können.
<i>Ästhetik und Minimalismus</i>	Dialoge sollten keine Informationen enthalten die selten oder irrelevant sind. Jede zusätzliche Einheit an Informationen sorgt für eine verminderte Sichtbarkeit der Dialoge. Ein Dialog mit nur relevanten Informationseinheiten ist leicht handhabbar.
<i>Unterstützung bei Fehlern</i>	Dem Benutzer soll bei Fehlern geholfen werden, indem Fehlermeldungen im Klartext (kein Code) angegeben werden und gleichzeitig eine konstruktive Lösung angeboten wird. Eine gute Fehlermeldung ist, erst keine aufkommen zu lassen.
<i>Hilfe und Dokumentation</i>	Eine gute Oberfläche sollte ohne Hilfe auskommen. Falls Hilfe dennoch notwendig ist, sollen sie den Benutzer bei der Bedienung des Systems unterstützen.

Tabelle 2: Allgemeine Usability-Heuristiken

Designprinzipien von Windows 10

In Tabelle 3 werden die *fünf Designgrundlagen* von Microsoft dargestellt. Diese Richtlinie sind Vorgaben für das Designen von Apps (weitere Informationen können hier entnommen werden [10]).

Richtlinien	Beschreibung
<i>Stolz auf Handwerkskunst</i>	Detailverliebtheit, besonders bei oft verwendeten Funktionen. System für ein breites Publikum bereitstellen d.h. auch für Menschen mit Beeinträchtigungen.
<i>Schnell und Fließend</i>	Der Benutzer soll direkt mit dem System interagieren können was die Begeisterung zum System steigert. Das Design sollte für Touch als auch direkte Interaktion konzipiert sein. Gleichmäßige Übergänge beim Seitenwechsel gewährleisten.
<i>Authentisch Digital</i>	<ul style="list-style-type: none"> - Lebhaftes Farben benutzen - Typografie gut einsetzen
<i>Mit weniger mehr erreichen</i>	Design auf das wesentliche reduzieren z.B. Redundanzen in der Oberfläche vermeiden
<i>Win as one</i>	Verwendung von bereits bekannte Gestaltungselementen schafft beim Benutzer ein Gefühl der Wiedererkennung und der Vertrautheit.

Tabelle 3: Designprinzipien von Microsoft

2.4 Psychologische Grundlagen der Systementwicklung

Mit den folgenden Gestaltgesetzen, die Teil der Gestaltpsychologie sind, soll ein Verständnis geschaffen werden, wie das menschliche Gehirn Informationen verarbeitet. Das Wissen über die menschliche Wahrnehmung ist für die Gestaltung von Benutzeroberflächen von großer Bedeutung, damit ein optimales Design für die Zielgruppe entwickelt werden kann. Gestaltgesetze sind keine Gesetze im eigentlichen Sinne, sondern dienen lediglich der Gruppierung und Abgrenzung ohne weitere Hilfsmittel.

2.4.1 Visuelle Wahrnehmung

Visuelle Wahrnehmung ist, was man mit den Augen sehen und dem Gehirn verstehen kann. Das menschliche Auge wandelt Lichtsignale in Sinnesreize um. Es unterscheidet zwischen bereits bekannten und neuen Reizmustern. Bereits früher erlernte Muster werden erkannt, selbst wenn sie sich im Detail unterscheiden, d.h. Wahrnehmungen neigen zur Wiedererkennung bzw. zur Vereinfachung [11]. Im Folgenden werden die Gestaltgesetze dargestellt und erläutert [12].

Gesetz der Figur-Grund-Trennung

Jedes Bild wird in Figur und Grund getrennt, wobei die Figur eine größere Aufmerksamkeit erhält. Eine Figur wird wahrgenommen, wenn sie die Eigenschaften eines kleinen und geschlossenen Bereichs mit symmetrischen oder konvexen Formen hat.



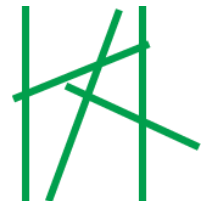
Gesetz der Nähe

Elemente, die räumlich nahe beieinander liegen, werden als Gruppe wahrgenommen [13]. Dadurch kann zum Beispiel auf Trennlinien und Rahmen verzichtet werden.



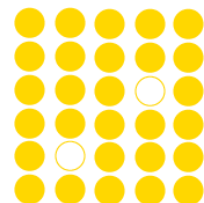
Gesetz der Symmetrie

Auf symmetrische, zueinander-stehende Elemente wird die Aufmerksamkeit schneller gelenkt, während zufällig angeordnete Elemente in den Hintergrund geraten. Damit kann die Aufmerksamkeit gelenkt werden.



Gesetz der Prägnanz

Unter gleichen Objekten werden zuerst die Wahrgenommen, die sich durch andere Eigenschaften von Rest abheben. So kann wichtige Information dargestellt werden.



Gesetz der Ähnlichkeit

Elemente die in Farbe, Kontrast, Größe und Form vergleichbar sind, werden gemeinsam wahrgenommen und zu einer Gruppe zusammengefasst.



2.4.2 Wirkung von Farbe

Farben spielen eine wichtige Rolle in unserem täglichen Leben. Jede Farbe hat eine bestimmte Wirkung auf den Menschen und löst verschiedene Reaktionen aus. Farben können Nachrichten übermitteln, deshalb muss man sich von vornherein im Klaren sein, welche Zielgruppe bzw. Kulturkreis damit angesprochen werden soll. Denn z.B. steht in westlichen Kulturen die Farbe Weiß für Reinheit, Klarheit oder Unschuld, wohingegen in östlichen Kulturen wie z.B. China Weiß als Symbol für die Trauer und den Tod gilt. Farben sind ein mächtiges Werkzeug für den Aufbau einer gelungenen Oberfläche. Die Farbumgebung eines Designs muss wohldefiniert und dokumentiert werden, damit Inkonsistenzen und unharmonische Kombinationen nicht entstehen können [6]. Bei Farben gilt das Prinzip *weniger ist mehr*, denn wenig Farben, die konsistent eingesetzt werden, führen letztendlich zu einem besseren Endergebnis [12]. In Tabelle 4 [12], [14] werden einige Farben und ihre Bedeutung erläutert.

Farbe	Bedeutung
Blau	Die wohl beliebteste Farbe steht für Vertrauen, Zuverlässigkeit, Sicherheit und Kraft. Blau erzeugt Ruhe und fördert die Konzentration und Kreativität. Blau wird als Informationsfarbe verwendet
Schwarz	Schwarz wird als edel, sachlich, stark, funktional oder modern assoziiert. Professionelle Produkte werden oft in Schwarz gehalten.
Grau	Ausgeglichenheit, Neutral oder Stille, Grau ist eines der häufigsten, beliebtesten und neutralen Töne. Grau funktioniert gut mit anderen Farben kann aber auch alleine verwendet werden.
Gelb	Die Farbe steht für Heiterkeit, gute Laune, Lebensfreude und Optimismus.

Orange	Orange strahlt Wärme aus und ist auffälliger als gelb. Orange wird gerne als Warnfarbe verwendet.
Weiß	Weiß steht für Reinheit und Sauberkeit.

Tabelle 4: Farben und ihre Bedeutung

3

Anforderungsanalyse

„If you don't know where you are going,
you are unlikely to end up there“
(Forrest Gump, Film 1994)

Der Fokus dieser Bachelorarbeit liegt vor allem auf den Phasen der Anforderungsanalyse und des User-Interface-Entwurfs (siehe Abbildung 1 [15]). Die Vorgehensweise der Arbeit bezieht sich auf das Referenzmodell der Daimler AG. Es ist ein Systementwicklungsprozess, das Usability Engineering in den Mittelpunkt stellt. Ziel des Modells ist ein System zu erstellen, das die Bedürfnisse der späteren Benutzer erfüllt. Der Vorteil dieses Modells im Vergleich zu anderen liegt darin, dass ergonomische Aspekte gleich zu Beginn mit einbezogen werden und eine fortlaufende Benutzerbeteiligung besteht. Alle Informationen betreffend des Referenzmodells stammen aus dem Skript „Usability Engineering“ WS13/14 [15]. Die festgestellten Ergebnisse werden im Folgenden näher erläutert.

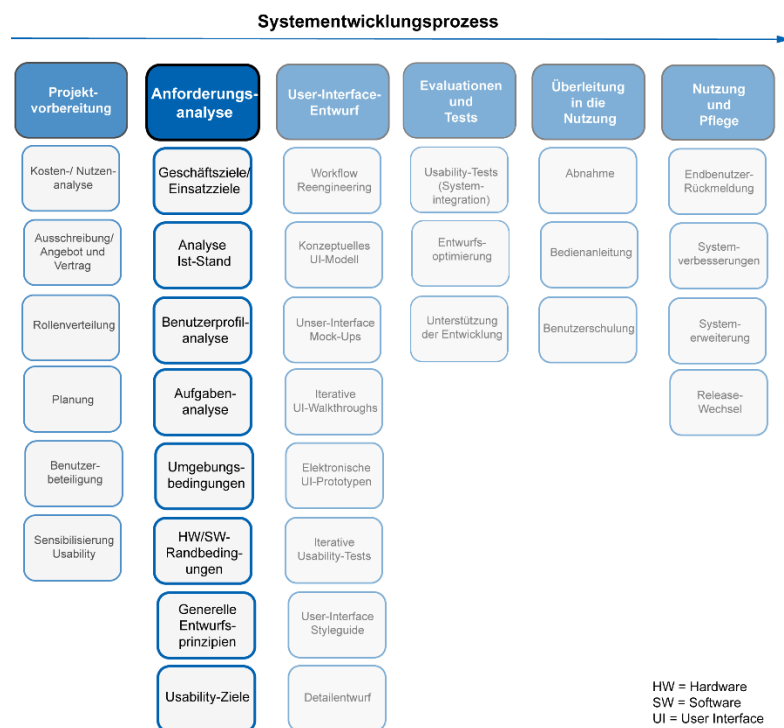


Abbildung 1: Anforderungsanalyse

3.1 Anforderungsanalyse

Die Anforderungsanalyse ist die wichtigste und entscheidendste Stufe im Systementwicklungsprozess. Hier wird die eigentliche Gestaltung des Gesamtsystems und der Dialoge festgelegt. Das Ziel der Anforderungsanalyse ist es, die wesentlichen Bestandteile, also was das System können muss, zu ermitteln. Dazu sind Informationen, Merkmale, Randbedingungen, Eigenschaften und Interessen der späteren Benutzer notwendig.

3.2 Geschäfts- und Einsatzziele

Als erster Schritt sollte so früh wie möglich die Analyse der Geschäfts- und Einsatzziele stattfinden, da sie harte Randbedingungen für das Design darstellen. „Aus diesen Zielen lassen sich die grundlegenden Designziele hinsichtlich der Aufgabenangemessenheit und Handhabbarkeit ableiten“ [15].

3.3 Analyse Ist-Stand

Dieser Schritt besteht aus der Analyse der Alt- oder Konkurrenzsysteme die sowohl positive als auch negative Aspekte beinhalten können. Aus diesen Erkenntnissen kann man das eigene System noch besser machen. In dem Fall von PHILharmonicFlows, existiert keine Modellierungsumgebung von Prozessen die sich so stark an der Datenstruktur orientiert. Demnach wurde auf eine umfangreiche Ist-Analyse verzichtet.

3.4 Kontextanalyse

Die Kontextanalyse ist der Oberbegriff, der in die Bereiche Benutzerprofil-, Aufgabenanalyse und Umgebungsbedingungen unterteilt werden kann, und bildet den dritten Schritt der Anforderungsanalyse. In dieser Phase geht es um die Analyse des Kontextes, in welche das System eingesetzt werden soll. Ziel von PHILharmonicFlows ist es, ein generisches System zu entwickeln, mit dem Prozesse aus unterschiedlichen Domänen abgebildet werden können.

3.5 Benutzerprofilanalyse

Im Bereich der Benutzerprofilanalyse werden die Eigenschaften des Endbenutzers untersucht und anhand dessen die softwareergonomischen Gütekriterien der Benutzerfreundlichkeit ermittelt. Es muss im Wesentlichen identifiziert werden, ob es sich, hinsichtlich der notwendigen Fachkenntnisse, um Laien oder Experten handelt. Es spielt also eine große Rolle, wie erfahren bzw. unerfahren die Zielgruppe in der Bedienung eines Computers ist und wie oft sie diese bei der Arbeit einsetzen. Ebenso wichtig ist

die Erfassung der Umgebungsbedingung der Umgebung in die das System eingesetzt wird. Die Oberfläche von PHILharmonicFlows dient in erster Linie als Demonstrationswerkzeug, daher richtet sich die Oberfläche an Benutzer aus dem akademischen Bereich der Informatik vor. Folglich liegen ihnen gute fachliche Kenntnisse vor und sie sind vertraut im Umgang mit Computern. Da die Oberfläche von PHILharmonicFlows überwiegend zum Demonstrationszweck genutzt wird, reichen die Ansicht eines Administrators und eines Endbenutzers. Welche Unterschiede die Benutzeroberfläche haben sollten, wird in Kapitel 6 näher erläutert.

3.6 Aufgabenanalyse

Bei der *Aufgabenanalyse* geht es darum, die Hauptaufgaben zu identifizieren, die der Benutzer im späteren System zu erledigen hat. Nach dem Referenzmodell der Daimler AG [15] sollte man wie folgt Vorgehen:

1. *Einbeziehung der Ergonomieexperten in die Gesamtsystemanalyse*
Diese können feststellen, ob alle wichtigen Funktionen die das neue System enthalten soll, beachtet worden sind.
2. *Analyse bereits vorliegender Anforderungen*
Vorhandene Altsysteme bilden eine gute, und vor allem kostenlose Quelle, um gewünschte bzw. erforderliche Anforderungen des Benutzers im neuen System zu realisieren.
3. *Brainstorming/Ideenfindung mit grafischer Visualisierung grober Abläufe*
Bei bereitgestellten Dokumenten vom Entwicklungsprojekt ist meistens ein hoher Anteil an Text vorhanden. Die darin vorhanden Anforderungen sollten für das Verständnis grafisch dargestellt werden.
4. *Definition von Funktionsbereichen und Betriebsabläufen*
Einfache Flussdiagramme sollen die groben Funktionalitäten und Abläufe des zu entwickelnden Systems beschreiben.
5. *Verfeinerung der Dialogsituationen und Anwendungsfälle mit Hilfe von Kontextfragen*
Ziel ist es eine detailliert Aussage für die späteren Entwurf der Benutzungsoberfläche zu erhalten.

Die Ergebnisse dieser Analyse sind in Kapitel 5 im Detail definiert.

3.7 Umgebungsbedingungen

In dieser Phase geht es um die Betrachtung der physikalischen Arbeitsumgebung des Systems. Bei PHILharmonicFlows beläuft sich der Standort auf ein Büro oder ein vergleichbares Umfeld, d.h. es müssen keine Schutzmaßnahmen gegen Vandalismus oder ähnliches beachtet werden. Arbeitsplätze müssen gewissen Normen einhalten, die in der *DIN 9241: Ergonomie der Mensch-System-Interaktion* festgehalten sind [16]. Unter anderem findet man dort eine Vorgabe eines hellen, freundlichen und lärmfreien Arbeitsplatzes. Dies gilt auch für die Oberfläche von PHILharmonicFlows. Zusätzlich sollte darauf geachtet werden dass der Hintergrund des Systems hell sein sollte um der Spiegelung der Oberfläche entgegen zu wirken.

3.8 Hardware- Software- Randbedingungen

Die Benutzeroberfläche ist von der Hardware- und Software- Randbedingung abhängig. Die PHILharmonicFlows Oberfläche wurde mit einer Bildschirmauflösung von 1280x720 Pixel gestaltet. Die Anwendung wird als Universal Windows-App implementiert, die auf jedem Gerät (z.B. PC, Tablet oder Smartphone) mit Windows 10 läuft. Die Idee hinter der Universal Windows-App ist, dass der Benutzer selbst entscheiden kann, auf welchem Gerät die Anwendung am einfachsten oder produktivsten einzusetzen ist [17]. Da sich die Arbeit nicht tiefer mit der Implementierung beschäftigt, wird auf diesen Punkt nicht detaillierter eingegangen.

3.9 Generelle Entwurfsprinzipien

Unter den generellen Entwurfsprinzipien versteht man die Normen, Richtlinien oder Styleguides, die bei der Systementwicklung berücksichtigt werden sollen. Diese Arbeit orientiert sich an den beschriebenen *Normen und Designrichtlinie* (siehe Kapitel 2.2) als auch an den *psychologischen Grundlagen der Systementwicklung* (siehe Kapitel 2.4). Es ist wichtig alle definierten Entwurfsprinzipien einzuhalten, um ein konsistentes und stimmiges System zu erhalten.

3.10 Usability-Ziele

Die Usability-Ziele beruhen auf den Ergebnissen der *Benutzerprofil-, kontextuellen Aufgabenanalyse* und den generellen *Geschäftszielen* [15]. Darauf basierend wird erfasst, welche Eigenschaften die Oberfläche von PHILharmonicFlows haben sollte.

- **Verständliche Informationsarchitektur**

Eine *verständliche Informationsarchitektur* schafft den Rahmen für eine gute Usability und kann grundlegend zur User Experience beisteuern. Aus diesem Grund sollen Informationen, welche für den Benutzer verfügbar sind, logisch strukturiert und leicht verständlich sein. Grundsätzlich sollten für den Benutzer nur die im aktuellen Kontext benötigten Informationen angezeigt werden und unwichtiges weggelassen werden. Beispielsweise kann diese durch eine Informationsfilterung geschaffen werden.

- **Verständliches Navigationskonzept [18]**

Das *Navigationskonzept* ist eines der wichtigsten Aspekte bei der Gestaltung einer Oberfläche. Eine gute Oberfläche sollte es dem Benutzer ermöglichen, durch alle Seiten leicht und verständlich navigieren zu können. Eine gut gestaltete Navigation sollte dem Benutzer helfen die folgenden Fragen zu beantworten:

- 1) *Wo befinde ich mich?*

Dem Benutzer muss zu jedem Zeitpunkt klar sein wo er sich gerade befindet.

- 2) *Wie bin ich hier hergekommen?*

Durch eine Pfadangabe kann verdeutlicht werden, wo sich der Benutzer befindet.

- 3) *Wohin kann ich von hier aus gehen?*

Durch Navigationshinweise kann dem Benutzer angezeigt werden, wohin er als nächstes gehen kann. Z.B. werden nur Icons oder ähnliches eingeblendet die tatsächlich ausgewählt werden können.

- 4) *Wie komme ich auf die Startseite zurück?*

Gerade bei tief verschachtelten Navigationsstrukturen ist es wichtig, schnell auf die Startseite zurück zu gelangen. Das bekannteste Navigationselement ist ein im Kopf der Seite angebrachtes Logo, das mit einem Klick zur Startseite führt.

- **Einfache Interaktion**

Durch ein gutes Interaktionskonzept kann die Konsistenz erhöht und die Interaktion vereinfacht werden. Es existiert bereits eine Vielzahl von Interaktionsgestaltungen, die sogenannten Interaktionsmuster. Interaktionsmuster beinhalten eine formalisierte Beschreibung elementarer Aktionen, die sogenannten Interaktionselemente, wie beispielsweise das Ausführen von Befehlen, das Menü,

den Button etc. Das Designprinzip für eine vereinfachte Interaktion ist unter *keep it simple and stupid* (KISS) bekannt [12].

- **Benutzerhilfe**

Für den Benutzer ist es wichtig stets über den Systemstatus informiert zu werden. Egal ob Anfänger oder Experte, jeder Benutzer sollte für ihn ausreichend Feedback und Hilfe vom System auf seine Aktionen erhalten. Um Informationen anbieten zu können, gibt es verschiedene Arten von Benutzerhilfen. Die *eingebetteten Hilfen*, darunter fallen Tooltips, Wasserzeichen, Eingebettete Hilfetexte und Assistenten, die *kontextuelle Hilfe* wie Hilfe-Panel, Einstiegsseiten oder Suchfunktionen oder die *separate Hilfe* mit einem Handbuch, Online-Hilfen oder Trainingsvideos.

- **Fehlermeldungen**

Fehlermeldungen sollten bei einem gut gestalteten Produkt erst gar nicht auftreten. Für Fehler, die dennoch auftreten, „soll eine klare Fehlerbeschreibung und eine Anleitung zur Fehlerbehebung angeboten werden“ [12].

3.11 Zusammenfassung

In diesem Kapitel geht es hauptsächlich um die Anforderung des Benutzers, d.h. wie und welche Aufgaben dieser im späteren System zu erledigen hat. Diese Phase stellt die Weichen für den nächsten Schritt, dem User Interface-Entwurf und bildet eine Grundlage für eine gute User Experience (siehe Kapitel 2.1.4)

4

User-Interface-Entwurf

„Eine Benutzerschnittstelle ist nicht dann perfekt,
wenn nichts mehr hinzugefügt – sondern
wenn nichts mehr weggelassen werden kann.“
(Merksatz [12] S. 145, Christian Moser (2012))

Die im Kapitel 3 erarbeiteten Ergebnisse der Anforderungsanalyse stellen die Basis für den folgenden Teil, dem User-Interface-Entwurf dar. Ziel des User-Interface-Entwurfs ist es, die erarbeiteten Anforderungen an das System in eine „bildlich sichtbare“ [15] Benutzeroberfläche umzusetzen. Ein gut entwickeltes Interaktionsdesign trägt maßgeblich zu einer guten Usability bei wie es schon im Kapitel 2 beschrieben worden ist. Es soll im Grunde eine „aufgabenangemessene Benutzungsschnittstelle entworfen werden“ [15], die sich an den zuvor definierten Usability-Zielen hält.

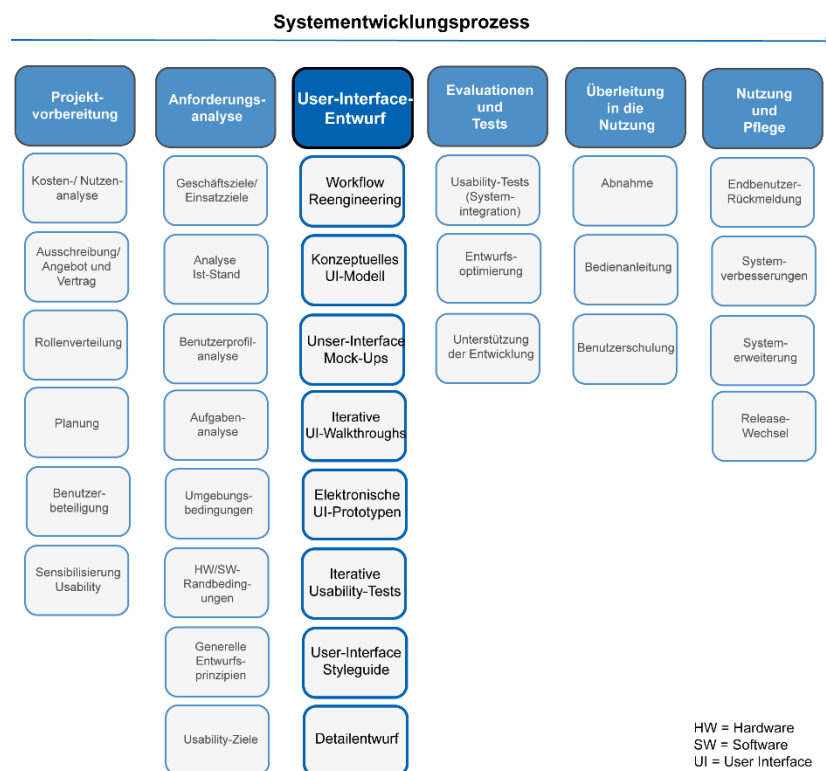


Abbildung 2: User-Interface-Entwurf

4.2 Workflow Reengineering

Der Startpunkt in der User-Interface-Entwurfs Phase ist das Workflow Reengineering. In diesem Schritt werden die Arbeitsabläufe der Benutzer analysiert und betrachtet in wie weit diese geändert wurden. Da noch keinen konkreten Endbenutzer existieren, wird dieser Schritt nicht berücksichtigt.

4.3 Konzeptuelles User-Interface-Modell

Die Basis des nächsten Schrittes bildet die in der Anforderungsanalyse und wenn durchgeführt die im Workflow Reengineering gewonnenen Erkenntnisse. Ziel dieser Phase ist es, eine einheitliche Grundlage für den weiteren Entwicklungsprozess zu schaffen. Es werden auf abstrakter Ebene Entscheidungen hinsichtlich des konzeptuellen Modells getroffen. Um dies sichtbar zu machen, werden User-Interface-Mock-Ups zu Hilfe genommen. Daher ist diese Phase eng mit der nächsten verbunden.

4.4 User-Interface Mock-Ups

Sobald die Grundlagen des User-Interface-Modells definiert sind, wird mit Hilfe der Mock-Ups die äußere Form des Systems evaluiert. So kann früh erkannt werden ob die Funktionen und grundsätzlichen Anordnung des Systems zur Erledigung der Aufgaben und Abläufe des Benutzers-, geeignet ist. Mit UI-Mock-Ups wird nicht das Design, sondern eine „Untermenge der Gesamtfunktionalität“ [15] betrachtet. Folgende Aktivitäten sollten nach dem Referenzmodell der Daimler AG [15] bei der Erstellung der UI-Mock-Ups betrachtet werden:

1. *Funktionalität auswählen*

Nur wichtige Dialoge (die beispielweise von allen Benutzern gebraucht werden), neue Eigenschaften, arbeitskritische Ergebnisse etc. in der Funktionalität testen.

2. *UI-Entwurf Skizzieren*

Grundsätzlichen Zweck des Dialogs veranschaulichen (aber nicht bis ins kleinste Detail).

3. *Mock-Ups erstellen*

Entweder Papier-Mock-Ups oder elektronisch laufende Prototypen.

4.5 Iterative User-Interface-Walkthroughs

Hauptziel dieser Phase ist, den Grobentwurf der Dialoge mit Hilfe der Endbenutzer zu evaluieren. Da noch keine eindeutigen Endbenutzer vorhanden sind, repräsentiert der Kunde den Endbenutzer, da er bestens mit den Funktionalitäten vertraut ist. Die Entwürfe wurden gemeinsam evaluiert, Probleme aufgeklärt und Lösungsansätze erschaffen. Da sich PHILharmonicFlows in der Entwicklungsphase befindet, können sich Anforderungen ändern bzw. stets neue hinzukommen.

4.6 Elektronische User-Interface-Prototypen

In diesem Schritt werden die gesammelten Ergebnisse in digitale Form gebracht um ein anschaulichen Design-Entwurf anzustreben. Die Umsetzung erfolgt mit Hilfe des Bildbearbeitungsprogramm Adobe Illustrator CS5. In dieser Phase können immer noch Änderungen und Fehler aufgedeckt werden.

4.7 Iterative Usability-Tests

Iterative Usability-Tests werden analog zu den iterativen UI-Walkthroughs behandelt. Der Unterschied hierbei ist, dass die zuvor erstellten elektronischen Prototypen getestet werden. Die Iterative-Usability-Tests wurden ebenfalls regelmäßig evaluiert.

4.8 User-Interface Styleguide

Sinn von User-Interface-Styleguides ist, einen Satz von Gestaltungsregeln festzulegen und beim Entwurf darauf zu achten das diese auch eingehalten werden. Sinn und Zweck ist, dass alle Dialoge einheitlich und konsistent gestaltet werden. Das Usability-Konzept wird in Kapitel 6 genauer erläutert.

4.9 Detailentwurf

Im letzten Schritt sollten alle Dialogbereiche und Dialogelemente im Detail festgelegt worden sein und falls notwendig um fehlende Details erweitert werden. Es ist wichtig, dass alle Konzepte einheitlich festgelegt sind. Die Ergebnisse des User-Interface-Entwurfs werden mithilfe von Screenshots in Kapitel 6 veranschaulicht.

4.10 Zusammenfassung

Ziel der Arbeit war, eine Oberfläche der Laufzeitumgebung von PHILharmonicFlows, unter Beachtung und Verwendung der in Kapitel 2 beschriebenen Usability Kenntnisse zu entwickeln. Die Oberfläche soll den Benutzer bei seinen Aufgaben unterstützen, einfach und intuitiv bedienbar sein und mit wenigen Klicks durch die Oberfläche navigieren lassen. Dies wirkt sich positiv auf die Zufriedenheit der Benutzer im Umgang mit dem System aus. Die gewonnenen Erkenntnisse aus den *Grundlagen der Usability* (siehe Kapitel 2), der *Anforderungsanalyse* (siehe Kapitel 3) und dem *User-Interface-Entwurf* (siehe Kapitel 4) liefern die Voraussetzungen für die Gestaltung der Oberfläche von PHILharmonicFlows. Alle Designentscheidungen hinsichtlich des visuellen Design, Interaktionsgestaltung und Navigationskonzepte werden im Detail in Kapitel 6 ausgearbeitet.

5

PHILharmonicFlows Konzept

*„Es geht nicht darum, dass man die richtigen Dinge tut,
sondern man muss die Dinge auch richtig tun.“*

P. Drucker (* 1909, † 2005)

Das Ziel von PHILharmonicFlows ist die Entwicklung eines datenzentrierten Prozess-Management-Systems. Herkömmliche Prozess-Management-Systeme laufen nach einem strengen Ablaufplan ab, der vor der Durchführung modelliert wird. Die Prozesslogik läuft somit nach festgelegten Arbeitsvorgängen ab. Zum Austausch von Anwendungsdaten innerhalb der Aktivitäten laufen Prozessflüsse zeitgleich zu den entwickelten Datenflüssen ab, der den Aktivitäten Ein- und Ausgabe-parameter zuweist [1]. Es ist daher sehr schwer während eines Arbeitsvorgangs Daten zu ändern. Am Beispiel des Bewerbungsprozesses kann dies veranschaulicht werden, denn es können sich jederzeit persönliche Daten des Bewerbers verändern. Ziel von PHILharmonicFlows ist es, den strengen Ablaufplan d.h. vor allem die Daten- und Prozesssicht transparenter zu gestalten, indem Änderungen während eines Arbeitsvorgangs leichter integrierbar sind [19]. Grundlage für die Ausarbeitung des Usability-Konzepts war die Modellierung der Laufzeitumgebung von *PHILharmonicFlows*. Wie in Abbildung 3 zu erkennen, ist PHILharmonicFlows in drei Strukturen gegliedert: die Datenstruktur, die Prozessstruktur und die Benutzerintegration. Um dieses Konzept besser verstehen zu können, werden im Folgenden die wesentlichen Bestandteile kurz erläutert.

5.1 Überblick über PHILharmonic Flows

PHILharmonicFlows stellt eine generische Prozessunterstützung für datenorientierte Prozesse zur Verfügung. Abbildung 3 fasst die Hauptkomponenten von PHILharmonicFlows zusammen [20]. Die Komponenten realisieren die Modellierung der Prozesssicht, Datensicht und die Verteilung der Benutzerechte. Zu Beginn muss das Datenmodell vom Benutzer modelliert werden. Es bildet das Fundament des Systems. Formulare, die für die Interaktion zwischen Benutzer und System erforderlich sind, werden dynamisch zur Laufzeit generiert. Darauf aufbauend werden die Prozesse erzeugt, die sich in Makro- und Mikroprozess aufgliedern. Zugriff auf Daten, sowie Rechte und Pflichten werden dem Benutzer über die Benutzerintegrations-Komponente zugewiesen.

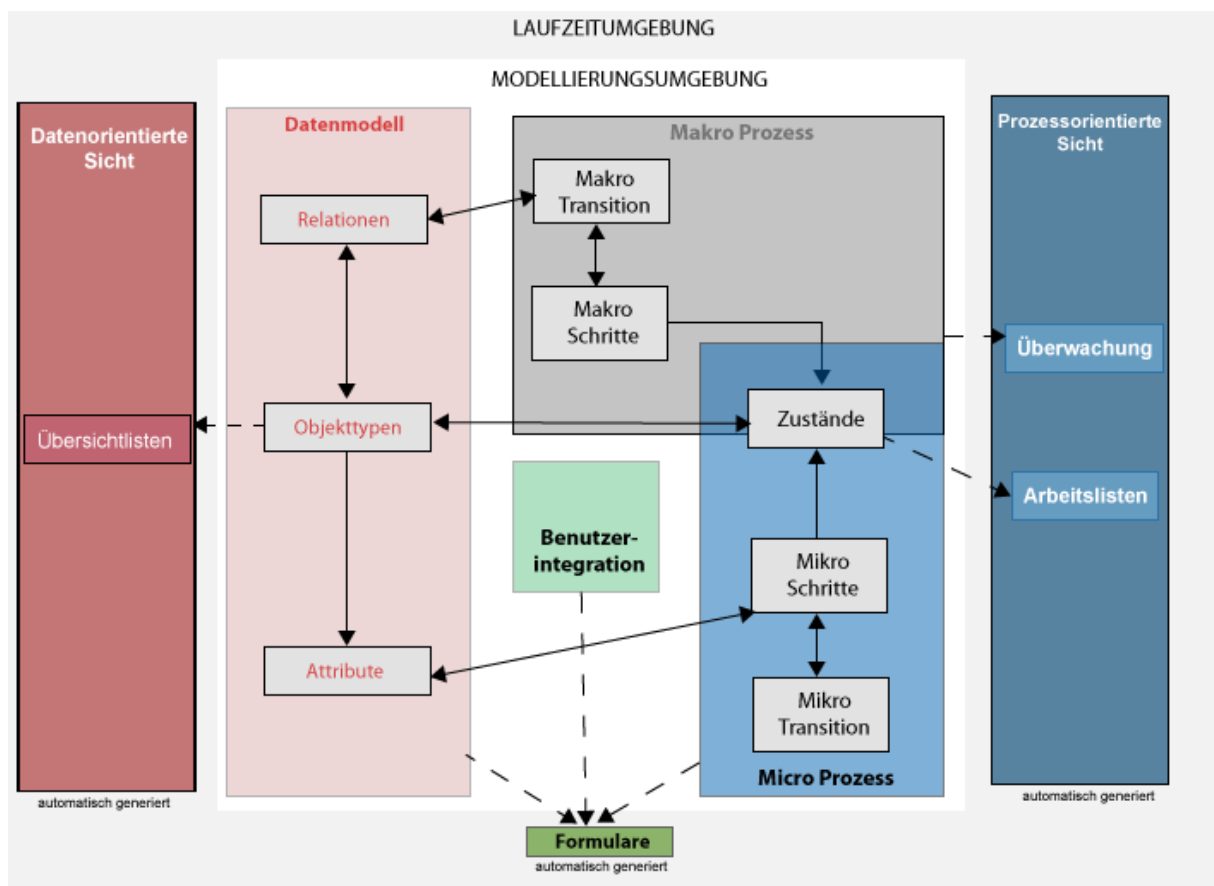


Abbildung 3: Die Hauptkomponente von PHILharmonicFlows

5.2 Datenstruktur

Die Datenstruktur legt fest, wie Daten in einem Prozess gespeichert und verknüpft sind. PHILharmonicFlows baut auf einer relationalen Datenbank auf, welche aus *Objekttypen* (object types), ihren *Attributen* (object attributes) und ihren *Relationen* (object relations) besteht.

5.2.1 Objekttypen

Ein Objekttyp dient als Container für die Daten. Dieser kann eine bestimmte Anzahl an Attribut-Typen beinhalten. Zur Laufzeit, werden für jeden Objekttyp beliebig viele Objektinstanzen erzeugt. Ein Objekttyp enthält mindestens ein Attribut. Ein Attribut beschreibt einzelne Eigenschaften des Objekttyps, besitzt einen eindeutigen Namen und einen eindeutigen Datentypen. Objektinstanzen unterscheiden sich in der Regel in ihren Attributwerten und den Relationen untereinander.

5.2.2 Relationen

Die Beziehungen zwischen den Objekttypen werden durch Relationen abgebildet. Die Relation beschreibt eine semantische Beziehung zwischen einer Quell- und einer Zielobjektinstanz. Für jede Relation kann eine maximale und minimale Kardinalität angegeben werden. Die Relationen zwischen Objekttypen werden mithilfe von Fremdschlüsseln modelliert. Automatisch wird durch die Relationen das Datenmodell in verschiedene Datenebenen aufgeteilt [20]. Auf Ebene 1 befinden sich Objekttypen, die keine anderen Objekttyp referenzieren. Die zweite Ebene referenziert den ersten Objekttyp und liegt auf zweiter Ebene. Objekttypen die diesen Typ referenzieren auf der dritten Ebene usw.

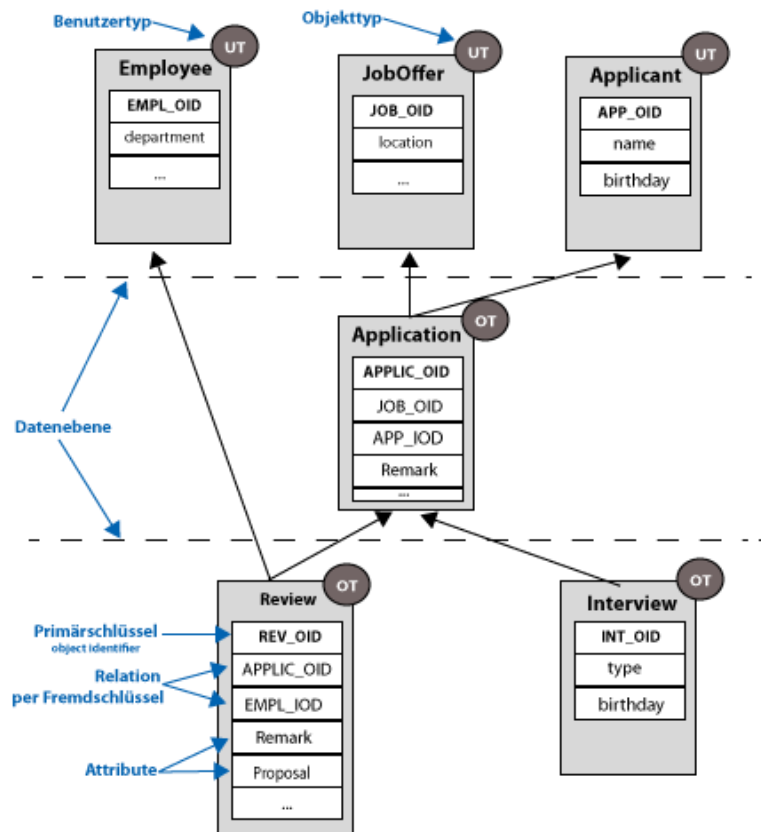


Abbildung 4: Beispiel für Datenstruktur

5.3 Prozessstruktur

Die Prozessstruktur legt den Ablauf des modellierten Prozesses fest. Man unterscheidet zwischen *Mikro-Prozessen* und *Makro-Prozessen*.

5.3.1 Mikro-Prozess

Jeder Objekttyp besitzt ein Mikro-Prozess, der das Objektverhalten definiert. Mikroprozesse bestehen aus *Mikro-Schritten*, *Mikro-Transitionen* und *Zuständen*. *Mikro-Schritte* legen Bedingungen fest, die zum Voranschreiten des Prozesses erforderlich sind. Mikro-Schritte sind Eingabefelder in Formularen. Jeder Mikro-Prozess Zustand besteht aus mindestens einem Mikro-Prozess-Schritt. Es gibt drei verschiedene Mikroprozess Schritte:

- *Leere Mikro-Schritte*, besitzen keine Bedingung und können zur Laufzeit immer erreicht werden.
- *Atomare Mikro-Schritte* referenzieren ein Attribut und kann erreicht werden, wenn dieser mit einem beliebigen Wert (ungleich null) belegt ist.

- *Wertespezifische Mikro-Schritte* referenzieren ein Attribut und haben zusätzlich mindestens einen festgelegten Wert. Sie können erreicht werden wenn das Attribut genau diesen Wert aufweist.

Der Ablauf der einzelnen Mikro-Schritte wird durch Mikro-Transitionen miteinander verbunden. Jeder Mikro-Prozess bildet einen zyklenfreien gerichteten Graph der von einem Startzustand in einen Endzustand führt. Man kann zwischen *internen* und *externen* Mikro-Transitionen unterscheiden:

- *Interne Mikro-Transitionen*, hier sind zwei Mikro-Schritte innerhalb eines gemeinsamen Mikro Zustandes verbunden.
- *Externe Mikro-Transitionen*, zwei Mikro-Schritte aus unterschiedlichen Mikro Zuständen verbunden.

In manchen Situationen ist es notwendig, in einen vorhergehenden Zustand zurückzuspringen zu können (Rücksprungtransitionen). Hier werden nicht zwei Mikro-Schritte verbunden, sondern Zustände. Jeder Mikro-Prozess verfügt über einen Startzustand und beliebig viele Endzustände. Mit Erreichen des Endzustandes ist der Mikro-Prozess beendet.

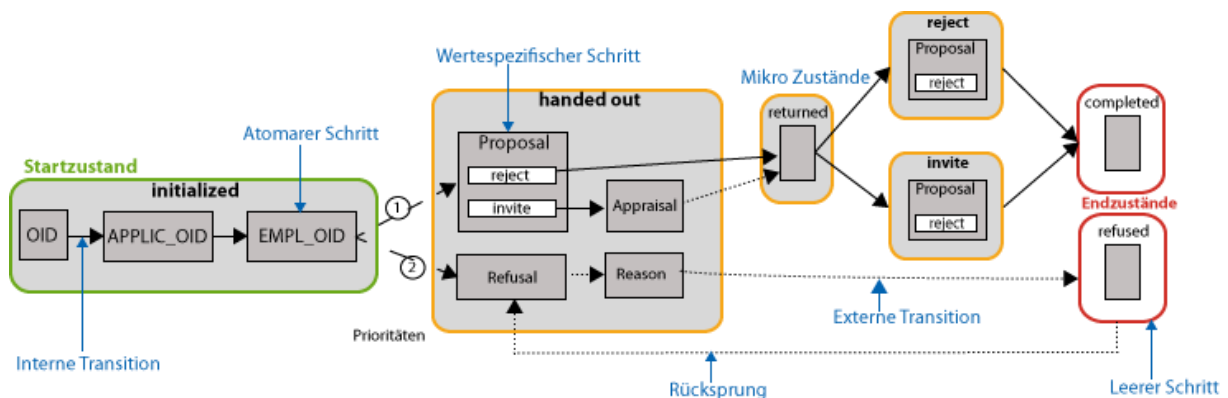


Abbildung 5: Beispiel Mikro-Prozess [20]

5.3.2 Zustände

Ein Mikro-Prozesstyp lenkt das Verhalten eines Objekttyps. Bei der Erzeugung einer neuen Objektinstanz wird automatisch eine neue Mikro-Prozessinstanz initiiert. Jede Mikro-Prozessinstanz besitzt unterschiedliche Laufzeitmarkierungen, die den Gesamtzustand abbildet. Für einen Zustand sind verschiedene Laufzeitmarkierungen vorhanden (Tabelle 5). Initial hat ein Zustand die Markierung WAITING, außer es ist ein Startzustand der als ACTIVATED markiert wird. Ein Markierungswechsel trifft ein, durch den Übergang einer Prozessinstanz in einen Folgezustand. Der aktivierte Zustand wird als CONFIRMED markiert und der Folgezustand als ACTIVATED. Sollten alternative Folgezustände existieren müssen diese als SKIPPED markiert werden, da immer nur ein Pfad des Mikro-Prozesses durchlaufen werden kann. Zustandswechsel werden erkannt, sobald externe Transitionen erreicht werden. Sobald die Laufzeitmarkierung auf READY gesetzt wird, wird der Vorgängerzustand beendet und der Nachfolgezustand aktiviert.

Zustandsmarkierung	Beschreibung
WAITING	Der Zustand wurde noch nicht aktiviert, kann aber zu einem späteren Zeitpunkt aktiviert werden.
ACTIVATED	Der Zustand ist aktiviert
CONFIRMED	Der Zustand ist abgeschlossen und der nächste Zustand wurde aktiviert.
SKIPPED	Der Zustand kann nicht mehr aktiviert werden, da er auf einem abgewählten Pfad der Ausführung liegt

Tabelle 5: Laufzeitmarkierungen eines Zustandes [21]

5.3.3 Mikroschritt

Verschiedene Markierungen sind nötig um den Prozesszustand der Mikroschritte anzuzeigen. Ein Mikroschritt stellt die Verbindung zwischen Daten und Mikro-Prozess dar. Wenn ein Wert für ein Attribut einer Objektinstanz eingegeben wird, kann der betreffende Schritt aktiviert und die Ausführung aktiviert werden. Das Verhalten eines Schrittes wird ebenso durch Laufzeitmarkierungen gelenkt. Insgesamt gibt es neun verschiedene Markierungen. Initial werden alle Schritte als WAITING markiert, außer es ist ein Startzustand dann wird dieser als READY markiert. Der erste Schritt im Zustand wird als ENABLED markiert, das bedeutet für diesen Schritt muss ein Wert

eingegeben werden, damit der Prozess fortfahren kann. Mögliche Markierungen der Mikroschrittinstanzen und Bedeutung werden in Tabelle 6 erklärt:

Schrittmarkierung	Beschreibung
WAITING	Der Zustand, in dem der Schritt liegt, wurde noch nicht aktiviert.
READY	Der Zustand wurde aktiviert, der Schritt kann im Moment aber noch nicht erreicht werden.
ENABLED	Der Schritt kann durch Eingabe eines Wertes aktiviert werden.
ACTIVATED	Der Schritt ist aktiviert. Ein Wert wurde eingegeben.
BLOCKED	Der eingegebene Wert muss verändert werden, da er in keinen Wertebereich fällt, der durch Werteschritte definiert wurde.
UNCONFIRMED	Der Schritt wurde besucht, aber der dazugehörige Zustand wurde noch nicht komplett durchlaufen
COFIRMED	Der Zustand wurde verlassen und der Schritt wurde aktiviert.
BYPASSED	Der Schritt liegt in einem aktivierten Zustand, der Pfad auf dem er liegt wurde aber abgewählt.
SKIPPED	Der Schritt wurde nicht aktiviert.

Tabelle 6: Laufzeitmarkierungen eines Schritts [21]

5.3.4 Makro-Prozess

Der Makro-Prozess verbindet die Zustände verschiedener Objekttypen untereinander. Abhängigkeiten zwischen Zuständen verschiedener Objekttypen können mithilfe von Makro-Transitionen und Makro Aggregationen verbunden werden. Makro-Prozesse wirken sich auf den Fortschritt des Prozesses aus. Weiter Informationen können der Dissertation von Dr. Vera Künzle [20] entnommen werden. Im Rahmen dieser Arbeit wird auf diesen Unterpunkt nicht näher eingegangen, da dieser nicht relevant für die Ausarbeitung war.

Benutzerintegration

Nachdem das Datenmodell und die Prozesse festgelegt worden sind, fehlt noch die Benutzerintegration. Zugriffskontrollen schützen Daten vor unbefugten und unberechtigten Zugriffen. PHILharmonicFlows legt für jeden Benutzer bestimmte Benutzerrollen fest. Ein Benutzer ist eine Objektinstanz eines bestimmten Objekttyps, der als Benutzertyp markiert wird. Jeder Benutzertyp beschreibt automatisch eine Benutzerrolle. *Rechte* und *Verantwortlichkeiten* können verschiedenen Rollen zugeteilt werden. Rechte sind optional, d.h. sie können, müssen aber nicht ausgeführt werden. Im Gegensatz dazu sind Verantwortlichkeiten verpflichtende Aufgaben die der Benutzer ausführen muss. Rechtetabellen mit Lese- und Schreibrechte kann auf die Attribute der Objekttypen zugeordnet werden, wobei das Schreibrecht auch eine Leseberechtigung einschließt. Ebenso kann jeder Rolle zusätzlich das Löschen oder Ändern von Objektinstanzen zugewiesen werden.

In der Prozessdefinition wird bestimmt, welche Benutzer für die Durchführung bestimmter Operationen berechtigt sind. Auf Basis der Mikro-Prozesse werden die Verantwortlichkeiten bestimmt. Dies wird durch die Benutzerrollen ermittelt, die für die Zuweisung der geforderten Attributwerte in einem bestimmten Zustand verantwortlich sind. Es werde ebenso Verantwortliche für explizite Mikro-Transitionen und Rücksprung Transitionen definiert. Zustandsbezogene Berechtigungen sind nötig, um Benutzer während der Prozessausführung einen zeitgleichen Zugang zu den Anwendungsdaten zu bieten. In der Rechtetabelle wird bestimmt ob ein Attribut eines Objekttyps Leserechte, Schreibrechte oder keine Zugriffsrechte besitzt [22]. Die im Prozessverlauf nötigen Arbeitslisten und Formular, lassen sich durch die Unterstützung der Rechtetabelle und der Prozessdefinition automatisch zur Laufzeit generieren.

5.4 Zusammenfassung

Die beschriebenen Grundlagen sollen einen groben Überblick über die Komplexität des PHILharmonicFlows Konzept schaffen. Für die Gestaltung der einzelnen Ansichten der Oberfläche sind folgende nötig: die *Arbeitslisten* Ansicht, die *Objektverwaltungs* Ansicht (nur Administrator), die *Objektstruktur* Ansicht (nur Administrator) und die Ausführungs Ansicht. In diesem Kapitel wurden nur für diese Arbeit notwendigen Merkmale betrachtet. Ausführliche Informationen können der Dissertation von Dr. Vera Künzle [20] entnommen werden. Das folgende Kapitel 6 stellt die entwickelte Benutzeroberfläche vor.

6

Usability-Konzept

„Design setzt Inhalt voraus.

Design ohne Inhalt ist kein Design, sondern Dekoration.“

Jeffrey Zeldman (* 1955)

Dieses Kapitel beschreibt das entworfene Konzept der Benutzeroberfläche von PHILharmonicFlows. Es baut auf den gewonnen Erkenntnissen der *Anforderungsanalyse* (siehe Kapitel 3) und dem *User-Interface-Entwurf* (siehe Kapitel 4) auf.

Der Fokus der Gestaltung lag darin, die Benutzeroberfläche für einen Benutzer mit administrativen Funktionen zu gestalten, da das System zu Demonstrationszwecken genutzt wird. Der normale Endbenutzer wird nicht außer Acht gelassen. Die Funktionen die ihm zur Verfügung stehen werden an den entsprechenden Stellen veranschaulicht.

6.1 Allgemeiner Aufbau

Die Positionierung der Bedienelemente innerhalb der einzelnen Seiten der Anwendung wird ähnlich gewählt, wie man es bereits aus anderen Systemen, wie z.B. Dropbox, kennt. Diese Konvention ist für die Benutzeroberfläche von PHILharmonicFlows beibehalten, um dem Benutzer einen vertrauten Anblick zu bieten und ihm die Einstiegshürde zu erleichtern [12]. Das Grundgerüst der Benutzeroberfläche besteht aus so wenigen Bedienelementen wie möglich, aber so vielen wie nötig. Jede Anwendungsseite ist wie in Abbildung 6 aufgebaut, d.h. sie besteht immer aus einem *Zurück-Pfeil*, einem *Logo*, einer *Suchleiste*, einer *Sidebar* und einem *Inhaltsbereich*. Im Folgenden wird jede Anwendungsseite kurz erläutert. Das *Logo* befindet sich in der oberen linken Ecke, die einen Link darstellt um auf die Startseite zurückzukommen. Gleich daneben befindet sich der *Zurück-Pfeil*, der immer eine Ebene zurück auf die zuvor besuchte Seite navigiert. Ähnlich wie man es aus anderen Systemen (Microsoft Word, Google Chrome...) kennt. Die *Lokale Navigation* befindet sich oben im System, hier werden Buttons platziert, die bestimmte Aktionen auslösen, wie z.B. das Löschen oder Bearbeiten einer Instanz.

Oben rechts im Fenster befindet sich das *Suchfeld*, welches den Benutzer unterstützen soll bestimmte Dinge, wie z.B. Prozesse, schneller wiederzufinden. Auf der linken Seite erstreckt sich senkrecht die *Sidebar*, mit ihren Hauptnavigationselementen. Der Inhaltsbereich befindet sich in der Mitte des Fensters. Dort werden die Hauptinhalte abgebildet. Es wird auf eine Brotkrümelnavigation verzichtet, da es nicht viele Ebenen gibt. Die *aktuelle Position* ist an mehreren Stellen erkennbar:

- 1) Button in der Sidebar ist farblich hervorgehoben
- 2) die Überschrift im Inhaltsbereich befindet sich in jeder Ansicht oben links.

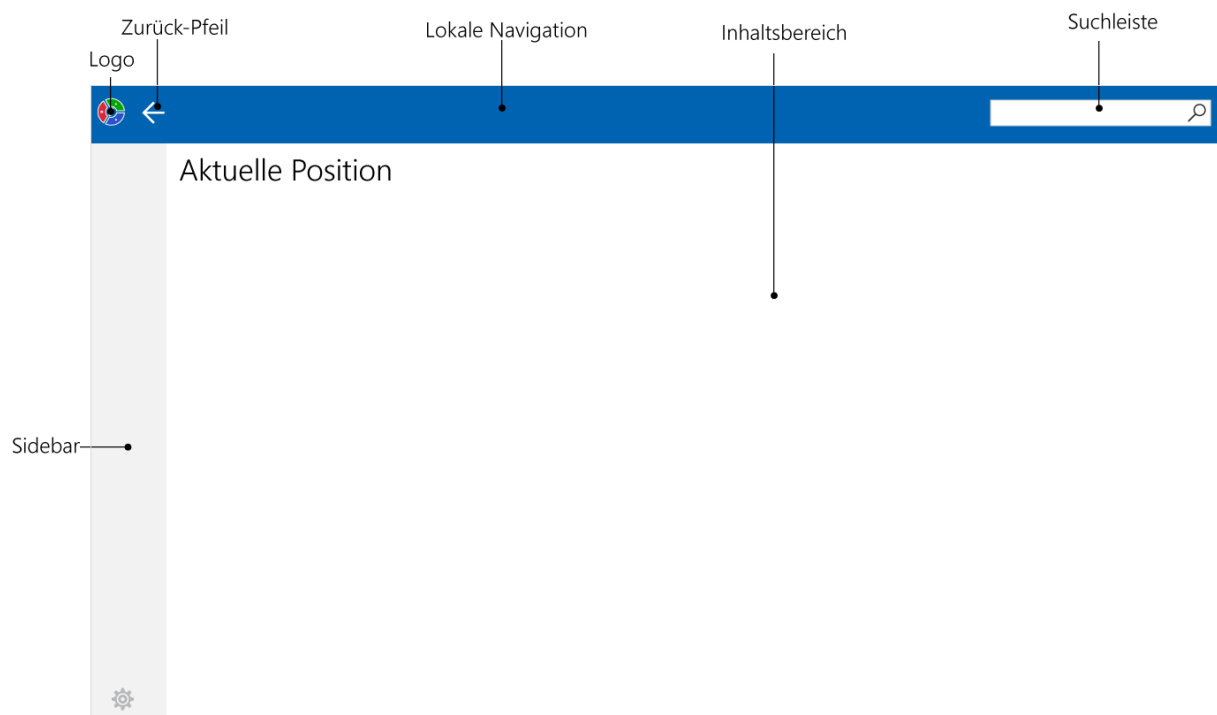


Abbildung 6: Generelle Bildschirmaufteilung

6.1.1 Die Sidebar

In der Sidebar sind die Hauptnavigationselemente untergebracht, die überall verfügbar sein sollten. Die Sidebar ist auf der linken Seite des Fensters platziert, weil es so den Lesefluss von links nach rechts unterstützt. Ebenso ermöglicht es, die wichtigsten Elemente auf einen Blick bereitzustellen, damit sich der Benutzer schneller zurechtfinden kann. Je nachdem, welche Rechte ein Benutzer hat, werden ihm mehr oder weniger Buttons in der Sidebar angezeigt. Man unterscheidet zwischen zwei verschiedenen Benutzern, dem Administrator und einen „normalen“ Endbenutzer. Die Sidebar eines Administrators besteht aus 5 Buttons (vgl. Abbildung 7). Die eines normalen Endbenutzers aus 3 Buttons (vgl. Abbildung 8). Nach kurzer Einarbeitungszeit sollten die Buttons für jeden Endbenutzer selbsterklärenden sein. Es wurde auf eine Beschriftung verzichtet, da Begriffe wie Objektverwaltung und Objektstruktur zu lang wären. Abbildung 7 und 8 zeigen die verwendeten Buttons (von oben nach unten wie folgt angeordnet):

Administrator Sicht:

- *Profileinstellung,*
- *Arbeitsliste,*
- *Objektverwaltung,*
- *Objektstruktur*
- *Ausführung.*

Normaler Endbenutzer

- *Profileinstellung,*
- *Arbeitsliste,*
- *Ausführung.*

Je nachdem welcher Button aktiviert wird, passt sich der Inhaltsbereich an. Im Folgenden werden die erstellten Benutzeroberflächen näher erläutert.

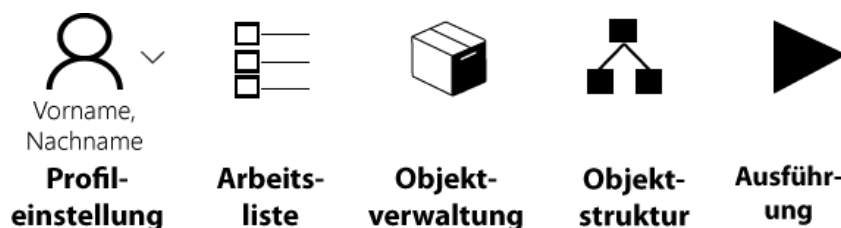


Abbildung 8: Sidebar Buttons (Administrator)



Abbildung 7: Sidebar Buttons (normaler Endbenutzer)

6.1.2 Anmeldung

Die Anmeldung erfolgt mittels Auswahl eines Projekts und Eingabe des Benutzernamens und Passwort (vgl. Abbildung 9). Durch ein Drop-Down-Menü kann ein Projekt ausgewählt werden. Für die Eingabe von Daten wie z.B. dem Benutzernamen und dem Passwort, wird ein Eingabefeld als Bedienelement verwendet. Ein Rahmen bündelt all diese Bedienelemente. Sollten dennoch Fehler bei der Texteingabe auftreten, wird dies visuell und deutlich kenntlich durch eine Fehlermeldung gemacht (siehe Kapitel 2).

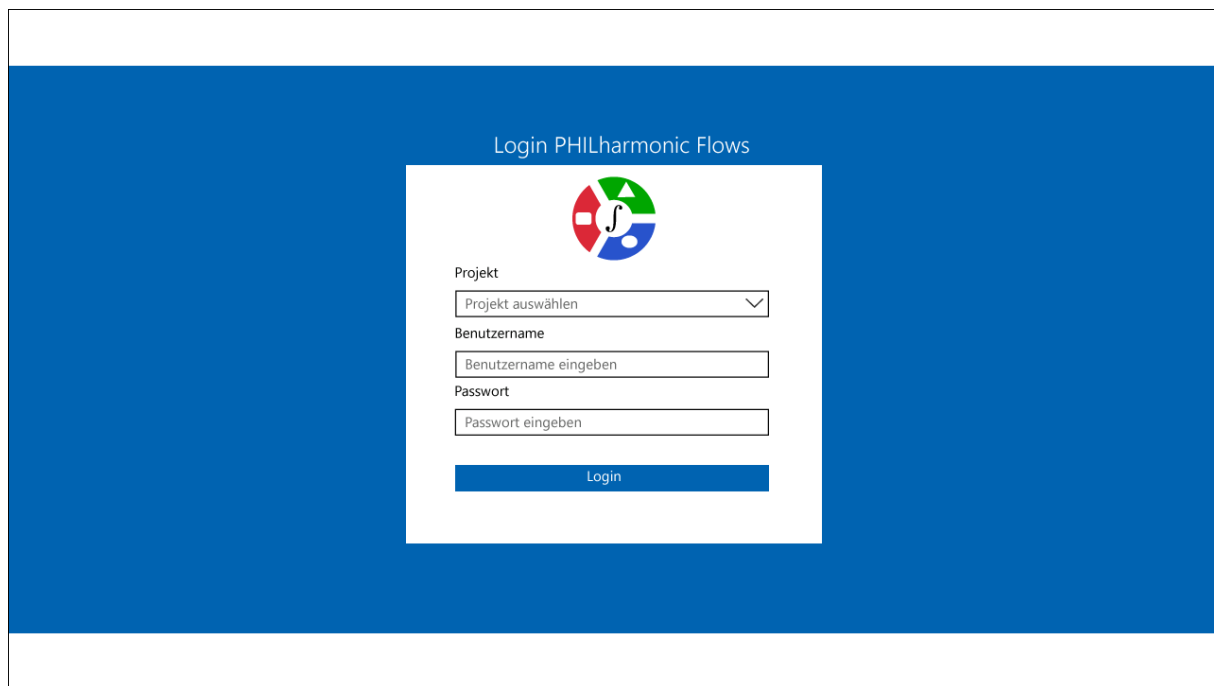


Abbildung 9: Anmeldung PHILharmonicFlows

6.1.3 Startbildschirm

Nach erfolgreichem Login zeigt sich der folgende Startbildschirm (vgl. Abbildung 11 und 12). Standardmäßig wird die Arbeitsliste angezeigt und der Arbeitslisten Button in der Sidebar ist eingefärbt (siehe Kapitel 6.2). Die Arbeitsliste zeigt dem Benutzer alle zu erledigenden Aufgaben an. Sie werden durch eine Tabelle dargestellt mit den Beschriftungen: Instanzname, Objekttyp Name, Aktueller Zustand, Datum und Status (siehe Abbildung 11). Insgesamt gibt es 4 verschiedene Statusinformationen, die mit Hilfe von eindeutigen Symbolen dargestellt werden und in Abbildung 10 erklärt sind.



Abbildung 10: Status Informationen

Je nachdem, welche Rechte ein Benutzer besitzt, sind unterschiedliche Schritte notwendig. Ein Administrator muss Objekte anlegen damit normale Benutzer sie bearbeiten können (das Anlegen von Objekten wird später erklärt). Ein normaler Benutzer wählt einen Prozess aus und wird zur Ansicht *Ausführung* geleitet.

Starten des Prozesses

In der Kopfzeile befinden sich 3 Schaltflächen: *Weiterleiten*, *Bearbeiten* und *Löschen*. Es gibt zwei Möglichkeiten einen Prozess auszuwählen um diesen zu bearbeiten:

- einen Doppelklick auf den Prozess startet die Prozessbearbeitung
- durch ein Klick auf den Radio-Button wird genau ein Prozess markiert und farblich hervorgehoben. Durch einen weiteren Klick auf die Schaltfläche *Bearbeiten* wird kann dieser Prozess bearbeitet werden.

Beide Möglichkeiten führen zu der Ansicht *Ausführung*

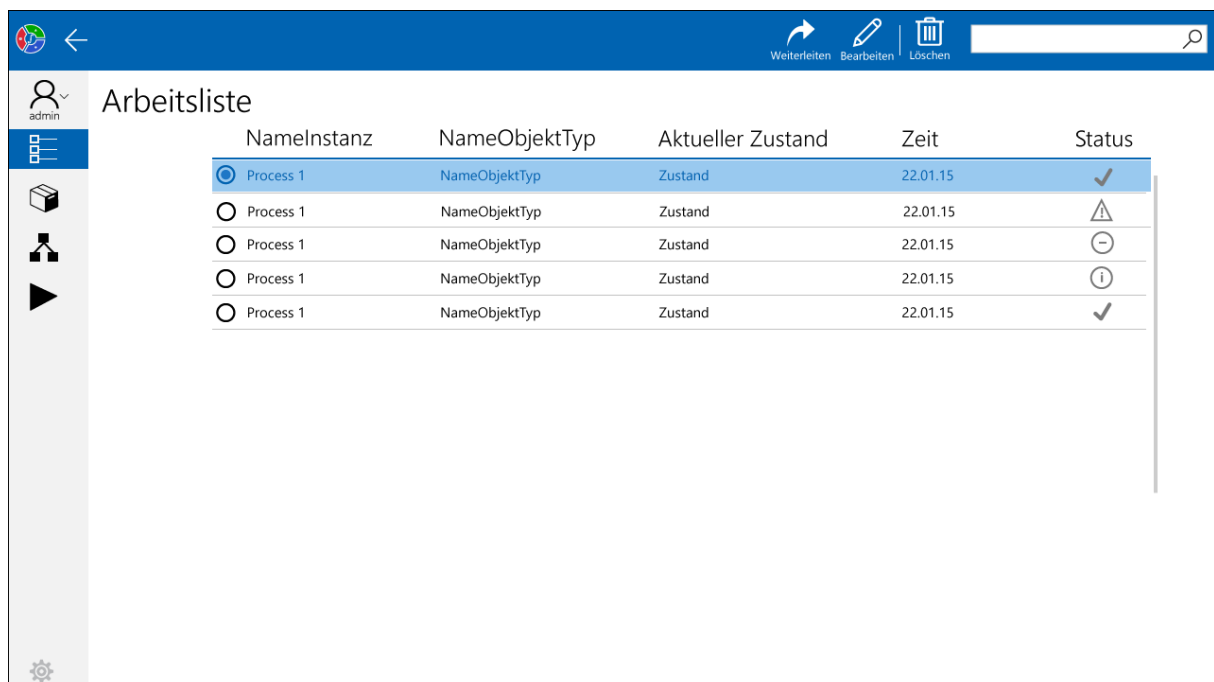


Abbildung 11: Startbildschirm Administrator Ansicht

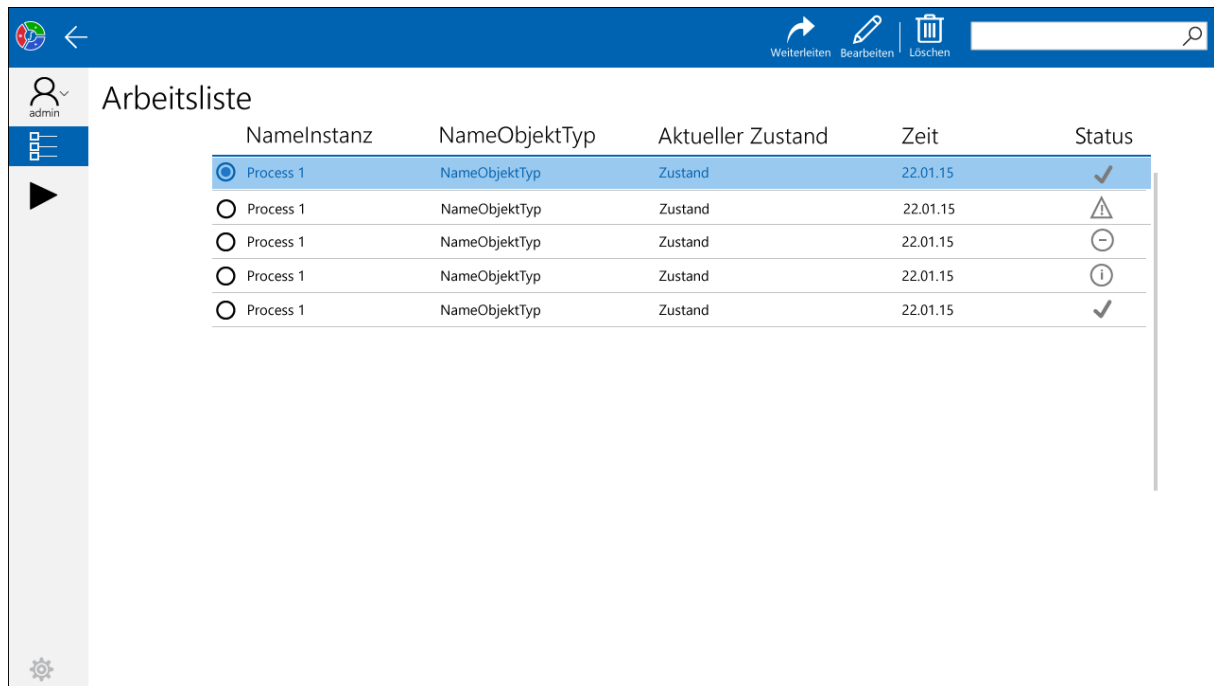


Abbildung 12: normaler Endbenutzer Ansicht

6.1.4 Ansicht Objektverwaltung

Die Ansicht der *Objektverwaltung* gibt autorisierten Benutzern einen Überblick über alle vorliegenden Objekttypen des aktuellen Projekts. Der Inhaltsbereich ist in zwei Bereiche aufgeteilt (vgl. Abbildung 15). Im linken Bereich werden die Eigenschaften der Instanzen mit dem Formular angezeigt. Im Rechten Teil sind alle Objekttypen in Kacheln gruppiert. Jede Kachel ist ähnlich aufgebaut. Im oberen linken Eck befindet sich der Titel des Objekttyps. Zu jedem Objekttyp können Objektinstanzen erzeugt werden. Das Erzeugen einer neuen Instanz ist über den Button *Instanzieren* im rechten Eck möglich. Die Objektinstanzen sind in einer Liste angeordnet und mit *Name*, *Zustand*, *Status*, *Geändert* und *Verknüpft* beschriftet. Das Scrollen durch die Objektinstanzen erfolgt vertikal. Eine Objektinstanz kann mit anderen verknüpft werden, was zur Laufzeit die Beziehungen zwischen den Objekten erstellt (vgl. Abbildung 14). Dies ist über den Button *Verknüpft* in der Kopfzeile möglich. Aufgrund dieser Beziehungen können instanzspezifische Rechte auf bestimmte Objektinstanzen vergeben werden. Beispielsweise kann eine Instanz des Objekts "Mitarbeiter" welcher mit einer Instanz eines Objekts "Stellenangebot" verknüpft wird auf alle Objekte "Bewerbung" welche wiederum mit dem Objekt "Stellenangebot" verknüpft sind, Schreibrechte bekommen (vgl. Abbildung 13). Diese komplexen Rechtebeziehungen werden zur Modellierzeit festgelegt (näheres kann der Arbeit von Dr. Vera Künzle [1] entnommen werden).

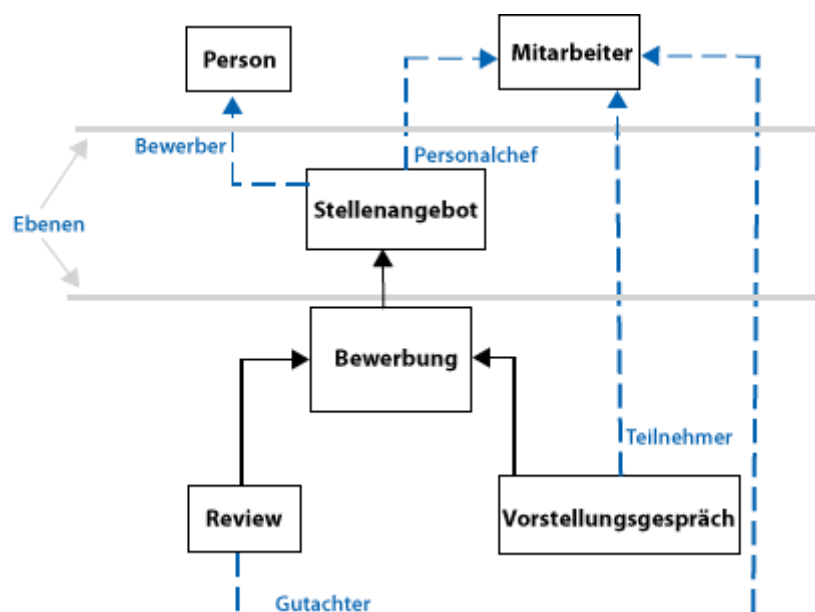


Abbildung: 13 Ebenen und Beziehungen

Objects

Name
Job Review

Zustand
State

Datum
22.01.2022

Urgency

Return date

Proposal

Reason

Alternative Job

Remark

Comment

Appraisal

Finished ☐

Job Review

Name	Zustand	Status	Geändert	Verknüpft
name	state	status	changed	linked
name	state	status	changed	linked
name	state	status	changed	linked
name	state	status	changed	linked

Application

Name	Zustand	Status	Geändert	Verknüpft
name	state	status	changed	linked
name	state	status	changed	linked
name	state	status	changed	linked

Job Offer

Name	Zustand	Status	Geändert	Verknüpft
------	---------	--------	----------	-----------

Interview

Name	Zustand	Status	Geändert	Verknüpft
name	state	status	changed	linked

Abbildung 15: Ansicht Objektverwaltung

Verknüpfe Job Review mit

Objects

Name
Job Review

Zustand
State

Datum
22.01.2022

Urgency

Return date

Proposal

Reason

Alternative Job

Remark

Comment

Appraisal

Finished ☐

Application

Name	Zustand	Status	Geändert	Verknüpft
name	state	status	changed	linked
name	state	status	changed	linked
name	state	status	changed	linked
name	state	status	changed	linked

Employee

Name	Zustand	Status	Geändert	Verknüpft
name	state	status	changed	linked
name	state	status	changed	linked
name	state	status	changed	linked

Abbildung 14: Objekte Verknüpfen

6.1.5 Ansicht Ausführung Administratorsicht

Nachdem Öffnen der Objektinstanz (durch Doppelklick der Instanz oder Auswählen der Instanz und Klick auf den Button *Bearbeiten*) wird die Benutzeroberfläche der Ausführung geöffnet. Die Überschrift im linken Eck des *Inhaltsbereiches* zeigt dem Benutzer welcher Prozess ausgewählt wurde. Durch einen Mausklick auf den kleinen Pfeil neben der Suchleiste klappt ein Drop-Down-Menü mit den möglichen Ansichten der Ausführung auf. Insgesamt kann zwischen sieben verschiedenen Ansichten ausgewählt werden (vgl. Abbildung 16):

- Alles (d.h. die Gesamtübersicht: Mikro- und Makroprozess + Formular),
- Makro- und Mikro-Prozess,
- Formular + Makro-Prozess,
- Formular + Mikro-Prozess,
- Makro-Prozess,
- Mikro-Prozess,
- Formular.



Abbildung 16: Ansichten Mikro-Prozess Darstellung

Der Micro Prozess kann auf zwei verschiedenen Arten ausgefüllt werden. Entweder über das Formular oder durch einen direkten Eintrag im Graphen. Ein Eintrag im Formular wird durch ein Formularfeld dargestellt und kann über den Button *OK* bestätigt oder den *Abbrechen* Button abgebrochen werden. Diese sind direkt über dem Formular in der Kopfzeile platziert. Je nach Rechtevergabe werden dem Benutzer nur die Formularfelder angezeigt, die ihm zugeordnet sind. Ein direkter Eintrag im Graph ist durch einen Klick auf das Formularfeld möglich. Es öffnet sich ein Pop-up-Fenster, indem die Werte eingegeben werden. Genaue Beschreibung zum Farbkonzept des Mikro-Prozess Graphen siehe Kapitel 6.2.

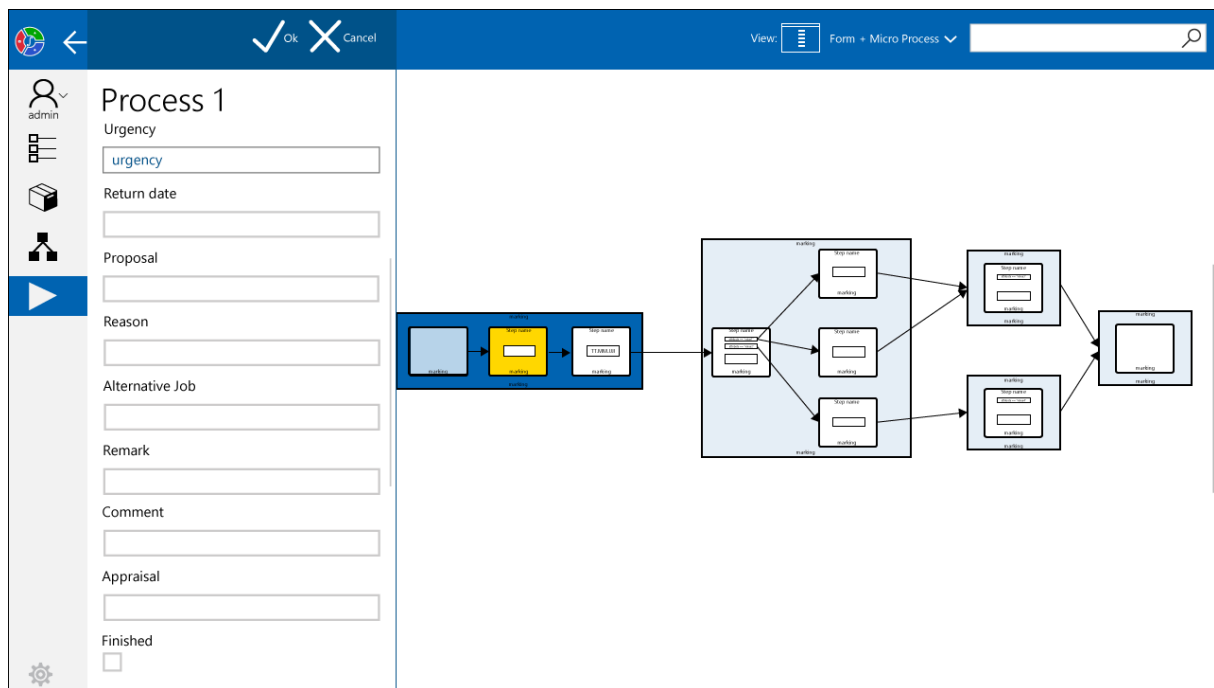


Abbildung 17: Ansicht Mikro-Prozess Darstellung

Ansicht Ausführung normale Endbenutzer-Sicht

Durch die Auswahl eines Prozesses aus der Arbeitsliste wird der normale Endbenutzer auf die Ansicht *Ausführung* weitergeleitet. Hier wird das Formular angezeigt, das vom Benutzer ausgefüllt und mit *Ok* bestätigt werden kann. Nach erfolgreicher Eingabe wird man zurück zur Arbeitsliste geleitet. Zu jeder Zeit kann der Vorgang mit dem Button *Abbrechen* abgebrochen werden.

The screenshot shows the 'Process 1' execution view. The left sidebar displays process details: Name Object Types, Name Object Type, State, Date, and 22.01.2022. The main area contains a form with the following fields: Urgency, Return date, Proposal, Reason, Alternative Job, Remark, Comment, Appraisal, and Finished. Each field has a corresponding input box or checkbox.

Abbildung 18: Prozess zur Bearbeitung ausgewählt

6.1.6 Ansicht Objektstruktur

In dem Menüpunkt Ansicht *Objektstruktur* (nur für Administratoren) werden die Beziehungen zwischen den Objekten grafisch dargestellt. Ein Objekt kann mehrere Instanzen besitzen, die in einem Kasten gruppiert werden. Jede Instanz besteht aus dem Objektnamen, Zustand und Status (vgl. Abbildung 19). Die Beziehungen werden durch Pfeile verdeutlicht. Die gruppierten Objekte können durch den Pfeilbutton minimiert als auch maximiert werden (vgl. Abbildung 20). In der *minimierten Ansicht* wird die Anzahl der Instanzen angezeigt (vgl. Abbildung 21)

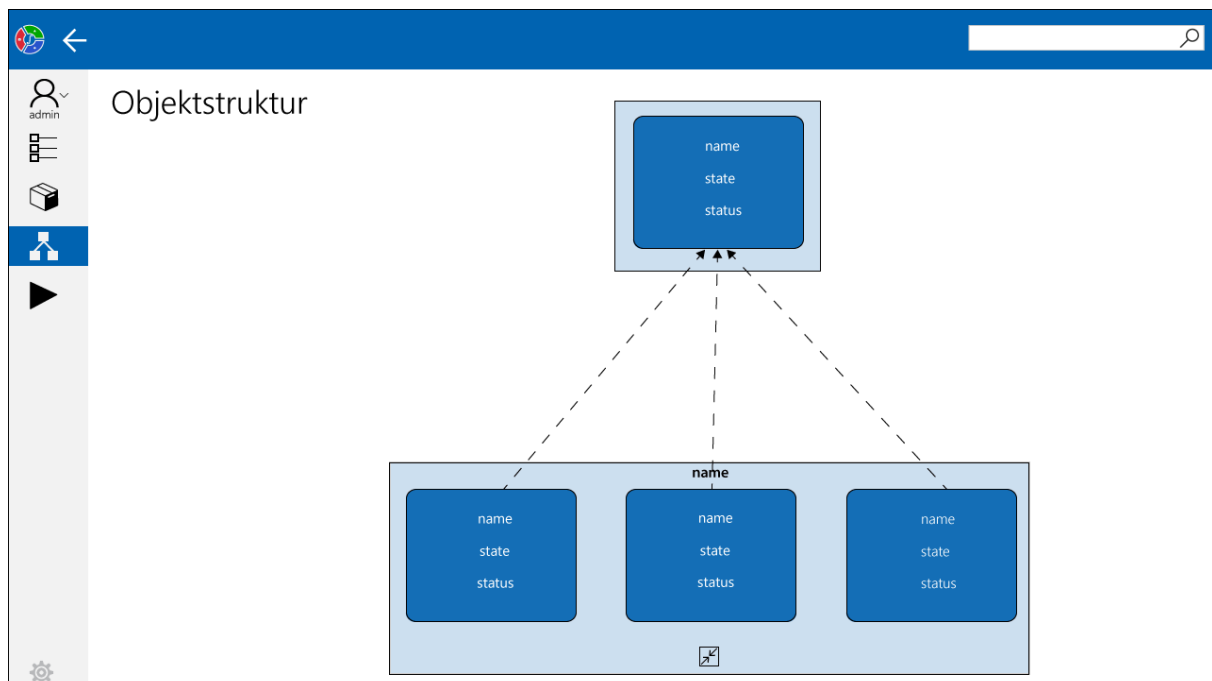


Abbildung 19: Baumstruktur

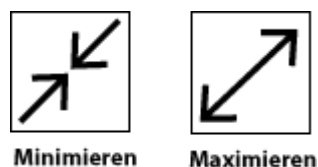


Abbildung 20: Button Minimieren und Maximieren

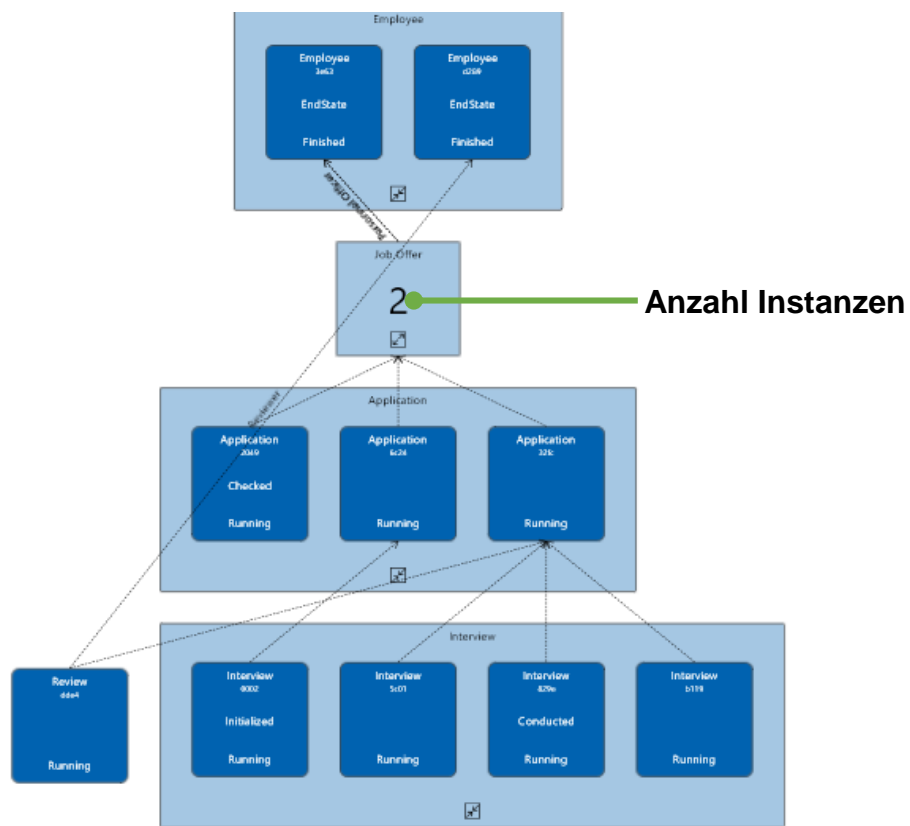


Abbildung 21: Komplexeres Beispiel eines Objektbaumes mit minimiertem Objekt

6.1.7 Der Mikro-Prozess Graph

Eine weitere Aufgabe dieser Arbeit war, den Graphen des Mikro-Prozesses neu zu gestalten. Das Farbbild des Graphen soll sich in das Design der Benutzeroberfläche integrieren. Darum wurde so gut es ging auf bunte Farben verzichtet. Um die Zustände und Schritte voneinander zu unterscheiden, wurde mit zwei Farben und Transparenz gearbeitet. Es ist wichtig, die verschiedenen Zustände und Schritte farblich hervorzuheben, um den aktuellen Zustand und den aktuellen Schritt zu erkennen. Die Flächen der Zustände wurden mit der Hauptfarbe der Benutzeroberfläche (dunkelblau RGB: 0/99/177) gestaltet. Je nach Zustand wird dieser nur in ihrer Transparenz verändert. Alle Zustände sind durch ein Rechteck mit schwarzem Rand dargestellt (vgl. Abbildung 23). Im Folgenden sind die jeweiligen Farben der Zustände näher erläutert.


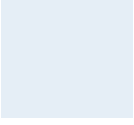
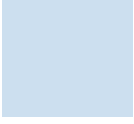

Zustandsmarkierung	Beschreibung
<i>ACTIVATED</i> (aktiviert) 	<i>ACTIVATED</i> wird mit 100%iger Deckkraft dargestellt. Da die anderen Zustände heller sind, wird der Fokus des Benutzers auf den aktivierten Zustand gelenkt (RGB: 0/99/177).
<i>WAITING</i> (noch nicht aktiviert) 	Der Zustand <i>WAITING</i> steht noch nicht im Mittelpunkt, darum wird dieser mit einer Deckkraft von 10% dargestellt. Dem Benutzer soll damit verdeutlicht werden, dass ein weiterer Zustand vorhanden ist, aber noch nicht aktiviert wurde bzw. nicht bearbeitet werden kann.
<i>CONFIRMED</i> (abgeschlossen) 	<i>CONFIRMED</i> hat die Deckkraft 20%. Es soll dem Benutzer verdeutlichen, dass der Zustand abgeschlossen ist, da es dunkler als <i>WAITING</i> aber heller als <i>ACTIVATED</i> ist.
<i>SKIPPED</i> (nicht mehr aktivierbar) 	Wie man es z.B. aus Microsoft Word kennt, werden Buttons die nicht aktivierbar sind grau hinterlegt. Darum wurde der Zustand <i>SKIPPED</i> in der Farbe Grau (RGB: 145/145/145) gewählt.

Tabelle 7: Zustandsmarkierung

Ein Schritt bzw. mehrere Schritte können sich in einem Zustand befinden und werden durch abgerundete Ecken dargestellt. Schritte werden mit dem Schrittnamen, der Werteeingabe und der Schrittmarkierung beschriftet (keine Werteeingabe bei einem Start- bzw. Endzustand). Wie anfangs erwähnt, wurde auf die Verwendung von viele Farben verzichtet. Es wird für die Zustände *CONFIRMED*, *UNCONFIRMED* die Hauptgrundfarbe dunkelblau (RGB 0/99/177) mit einer Transparenz von 20% verwendet.

Die *ENABLED*-Schritte werden in einem Gold, Gelb (RGB: 255/215/0) markiert, um den Blick des Benutzers auf den zu aktivierenden Schritt zu lenken.

Zustände wie *READY* und *WAITING* sind so lange weiß hinterlegt, bis der Zustand erreicht und ein Wert eingegeben werden kann. Mit dieser Farbe lässt sich gut verdeutlichen welcher Wert noch nicht eingegeben worden ist. *BYPASSED* und *SKIPPED* werden wie schon oben beschrieben in der Farbe Grau dargestellt.

Da die Farbe Rot (RGB 239/11/6) eine Signalfarbe ist, wird sie für den Zustand *BLOCKED*, also der fehlerhaften Eingabe eines Benutzers, verwendet (vgl. Kapitel 2).

Sollte aus irgendwelchen Gründen ein Schritt rückgängig gemacht werden, wird im Graphen und Formular ein orangefarbener (RGB: 246/142/51) Hinweis eingeblendet. Im Anhang A befindet sich ein Beispiel eines Mikro-Prozess Graphen, woran die Farbdarstellung veranschaulicht wird.

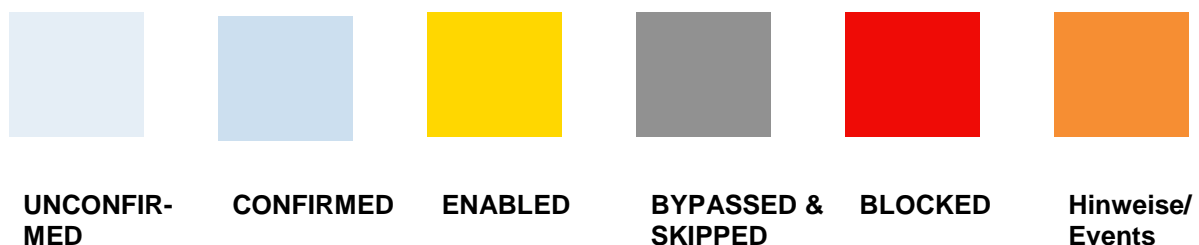


Abbildung 22: Farben Schritte Mikro-Prozess

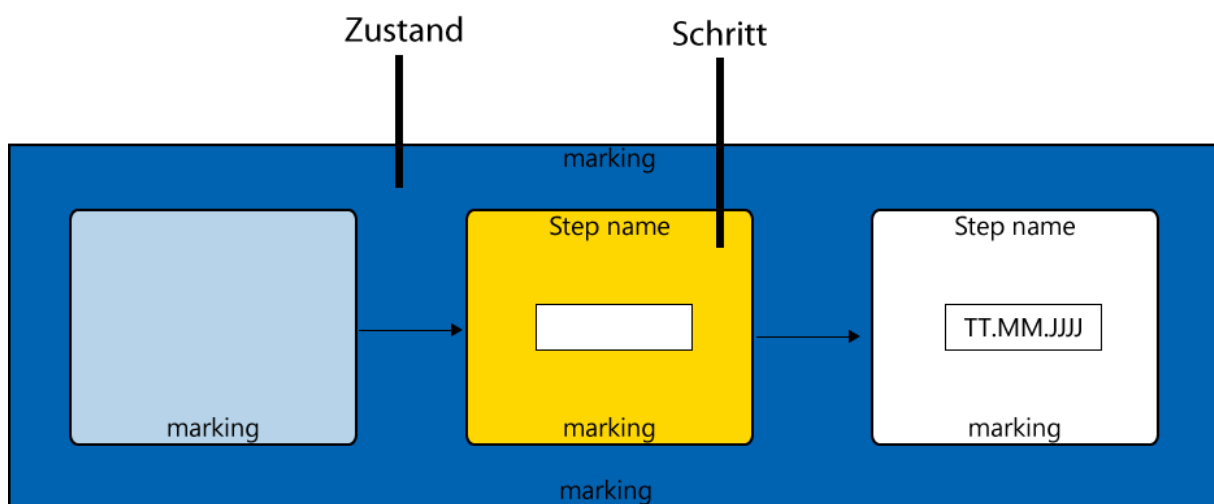


Abbildung 23: Ausschnitt Mikro-Prozess

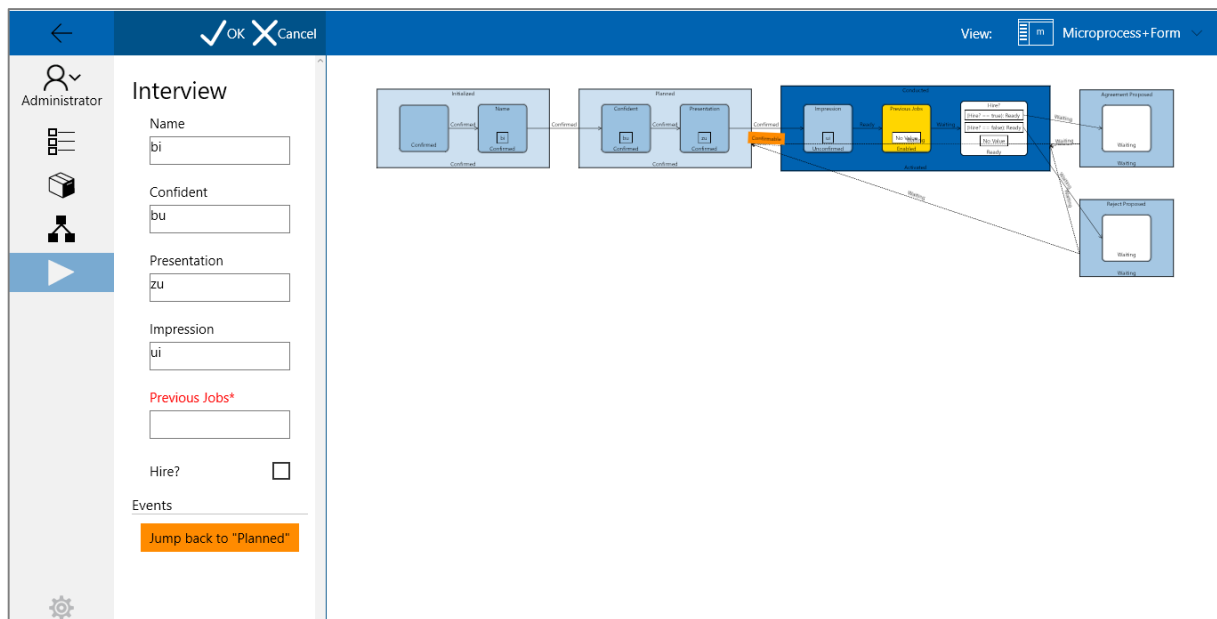


Abbildung 24 Hinweisdarstellung im Formular und Graphen

6.2 Farbkonzept

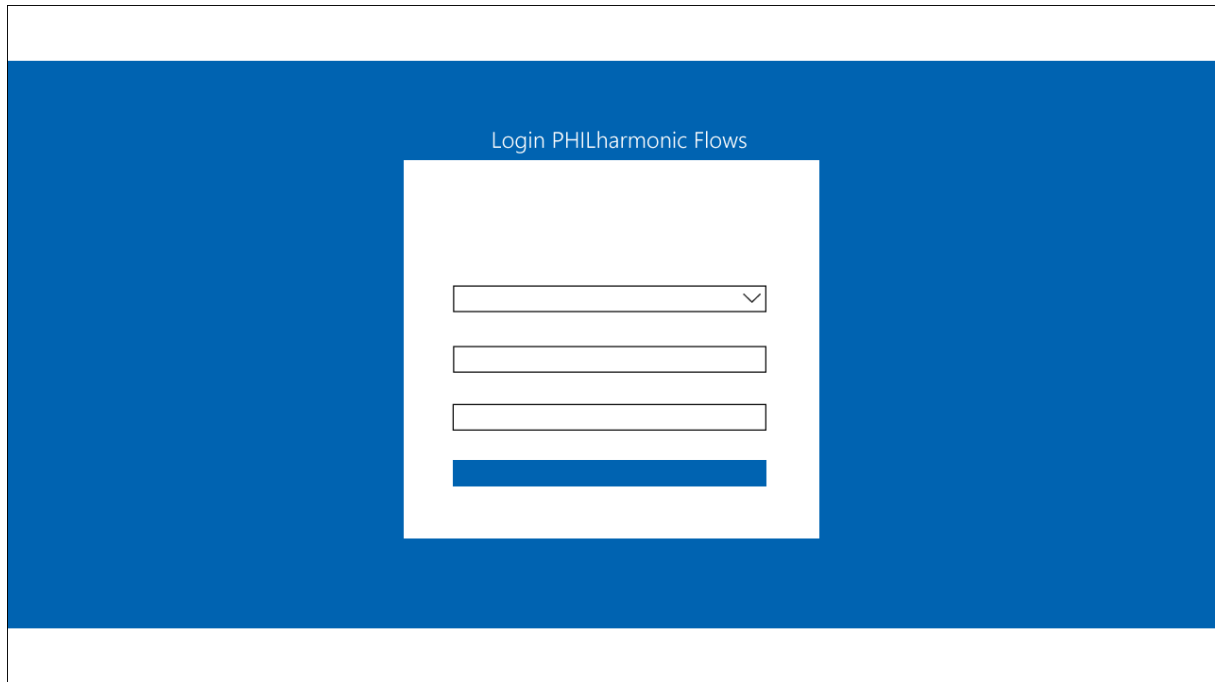
Farbe ist ein zentraler Bestandteil unseres Lebens. Menschen sehen und reagieren auf verschiedene Farben. Sie haben sowohl eine emotionale als auch psychologische Auswirkung. Sie ziehen die Aufmerksamkeit auf sich und beeinflussen den Menschen auf Grundlage ihrer Erfahrungen [12]. Die folgenden Farben sollen dem Benutzer bei seinen Aufgaben unterstützen und nicht ablenken (vgl. Kapitel 2).

Farbharmonie und Farbauswahl

Das Farbkonzept von PHILharmonicFlows wurde mit so wenigen Farben wie möglich gestaltet um den Benutzer nicht vom Wesentlichen abzulenken. Farben müssen sparsam, gezielt und konsequent verwendet werden. Es wurde daher eine Hauptfarbe gewählt, die durchgehend im Design verwendet wird. An gewissen Stellen wird mit ihrer Transparenz gearbeitet um im Farbschema zu bleiben. Die Hauptfarbe wird durch die Farbe dunkelblau(RGB: 0/99/177) dargestellt. Diese Farbe drückt Seriosität und Ruhe aus, was das schlichte Design unterstreichen soll (siehe Kapitel 2.4.2).

Abbildung 26 zeigt die Anmeldung am System, in dem ein Teil des Hintergrundes in der Hauptfarbe gestaltet ist. Abbildung 25 zeigt die Farben des Startbildschirms (nach der Anmeldung). Die Farbe Hellgrau (RGB: 242/242/242) ist für die Sidebar bestimmt. Es ist ein neutraler Ton und kommt in Kombination mit Blau gut zur Geltung. Die Hintergrundfarbe des Inhaltsbereiches ist in Weiß gehalten, damit die Schriftfarbe schwarz besser zur Geltung kommt. Ebenso wurde darauf geachtet, dass die Sidebar und die lokale Navigation klar abgegrenzt und erkennbar sind.

In Tabelle 8 sind alle verwendeten Farben der Benutzeroberfläche dargestellt.



The image shows a login interface for 'PHILharmonic Flows'. It features a central white rectangular form on a solid blue background. The form is titled 'Login PHILharmonic Flows' at the top. Below the title, there are four input fields: a dropdown menu with a small downward arrow icon, followed by three standard text input fields. At the bottom of the form is a solid blue rectangular button. The entire interface is framed by a thin white border at the top and bottom.

Abbildung 26: Farbwahl Anmeldung



Abbildung 25: Farbwahl Startseite


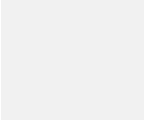
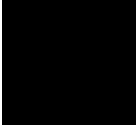
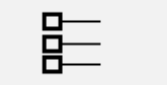
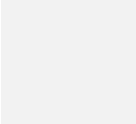

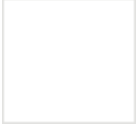



Erklärung	Farbe	Beispiel/ Beschreibung
Allgemein		
Hauptfarbe (RGB 0/99/177)		
Fläche der Sidebar (RBG: 242/242/242)		
Inaktiver Button		
Sidebar Aktiver Button		 (innen hellgrau RBG: 242/242/242) außen Hauptfarbe RGB 0/99/177)
Button lokale Navigation & Schriftfarbe (Weiß RGB: 255/255/255)		
Auswahl Instanz		Hauptfarbe mit Transparenz von 40%
Flächenhintergrund Button bei Benutzeroberflächenaufteilung (RGB: 0/82/137)		
Menütext (Schwarz RGB: 0/0/0) & (Weiß RGB: 255/255/255)	ABCabc	

Tabelle 8: Farben Allgemein

Die Kacheln der Objekte in der Ansicht *Objektverwaltung* sind in der Hauptgrundfarbe (RGB: 0/99/177) mit einer Transparenz von 10 % eingefärbt. Da es wichtig ist ein freundliches und schlichtes Farbbild zu erstellen wurde mit so wenigen Farben wie möglich gearbeitet. Wird eine Instanz ausgewählt, verfärbt sich der Rahmen der entsprechenden Kachel in der Hauptgrundfarbe (RGB: 0/99/177) und die Instanz mit einer Transparenz von 40% der Hauptgrundfarbe. Die Beschriftungen der Tabellen sind ebenfalls in der Hauptgrundfarbe dargestellt.

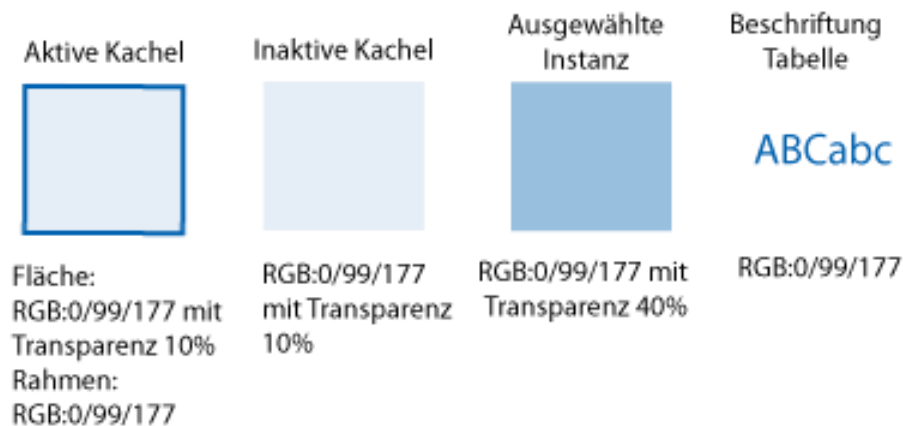


Abbildung 27: Objektverwaltung

Für die Ansicht *Verknüpfen* wurde eine analoge Farbe zu dunkelblau ausgewählt, nämlich türkisblau. Analoge Farben liegen auf dem Farbkreis direkt nebeneinander und wirken auf das Auge angenehm. Eine ausgewählte Instanz wird durch ein dunkleres türkisblau verdeutlicht (RGB: 3/131/135). Aktive Kacheln unterscheiden sich durch den schwarzen Rahmen von inaktiven Kacheln.

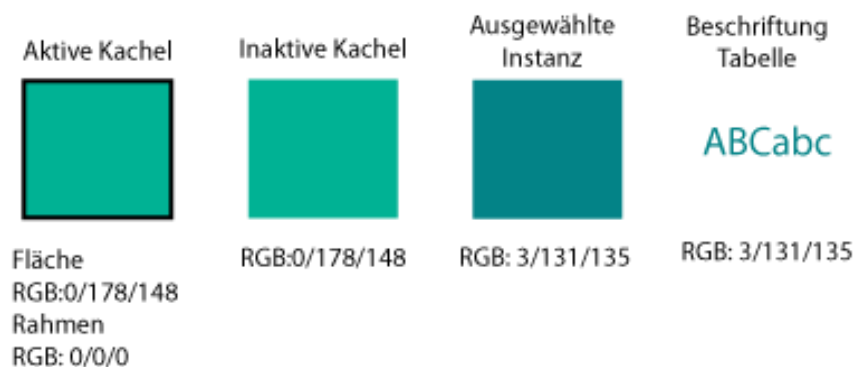
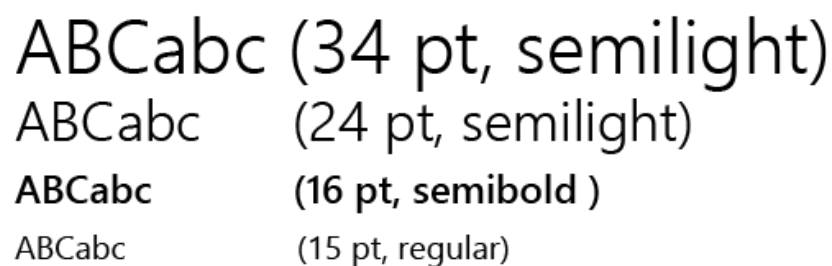


Abbildung 28: Verknüpfte Objektansicht

6.3 Schrift

Die ordnungsgemäße Verwendung von Schriftgrad, -breiten, -farben, Laufweite und Abstand kann dazu beitragen, dass die Benutzeroberfläche ein sauberes und übersichtliches Erscheinungsbild verleiht und damit die Verwendung erleichtert [17]. Da die Benutzeroberfläche von PHILharmonicFlows nach dem Windows 10 Prinzipien gestaltet worden ist, wird die Schriftart *Segoe UI* verwendet. Es ist eine von Microsoft entwickelte Schriftfamilie die serifenlos ist und sich gut für den Einsatz am Bildschirm eignet. Es werden unterschiedliche Schriftgrößen verwendet um die Benutzeroberfläche übersichtlich gliedern zu können. Die Schriftgröße für Überschriften werden in 34 Punkt semilight, Unterüberschriften in 24 Punkt semilight und Mengentexte je nach Verwendung in 16 Punkt semibold bzw. 15 Punkt regular dargestellt.



ABCabc	(34 pt, semilight)
ABCabc	(24 pt, semilight)
ABCabc	(16 pt, semibold)
ABCabc	(15 pt, regular)

Abbildung 29: Schriftgröße und -schnitt

6.4 Symbolik

Ein Icon, ist ein Piktogramm, das Informationen in einfachen Bilder bzw. Zeichen darstellt. Im Vergleich zu Texten haben Symbole viele Vorteile, sie sind sprachneutral, leicht einprägsam und brauchen weniger Platz auf der Benutzeroberfläche [12]. Bei Icons muss darauf geachtet werden, dass sie für den Benutzer verständlich, einfach und prägnant sind. Deshalb werden die meisten Icons aus geometrischen Grundformen erstellt. Die verwendeten Icons orientieren sich an der Windows 10 Symbol-sammlung. Icons sollten den Benutzer mit einfachen und flexiblen Wegen zum Ziel seiner Aufgabe führen. In der Sidebar wurde auf Beschriftungen verzichtet, da Begriffe wie Objektverwaltung und Objektstruktur zu lang wären. Die Icons sind nach kurzer Einarbeitungszeit selbsterklärend (vgl. Abbildung 30).

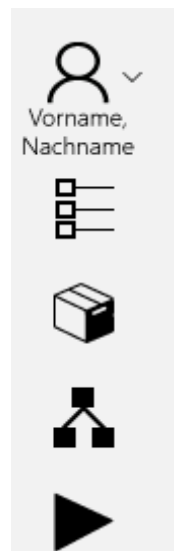


Abbildung 30: Ausschnitt Sidebar

Abbildung 31 zeigt die Icons, die in der lokalen Navigation verwendet werden. Icons die bestimmte Aktionen ausführen, werden durch bereits bekannte Symbole dargestellt, wie z.B. ein Icon für Ok, um die Eingaben eines Prozesses zu bestätigen.



Abbildung 31: Icons die Aktionen auslösen

Das *Ansichten Icon* (siehe Abbildung 32) beschreibt textuell als auch grafisch die jeweiligen Ansichten.



Abbildung 32: Ansichten Mikro-Prozess Darstellung

Der Status der Prozesse wird durch einfache Symbole dargestellt (vgl. Abbildung 33). In den regelmäßigen Treffen mit dem Kunden wurde entschieden, dass hierfür keine Farben verwendet werden. Da das Design der Benutzeroberflächen schlicht wirken und nicht durch bunte Farben vom wesentlichen ablenken soll.



Abbildung 33: Statusinformationen

Die Minimieren- bzw. Maximierenfunktion ist mit Pfeilen dargestellt (vgl. Abbildung 34). Die Pfeilspitzen bestimmten dabei die Richtung der Verkleinerung bzw. Vergrößerung der Objekte. Zeigen die Pfeilspitzen aufeinander, wird durch einen Klick das Fenster minimiert und die Anzahl der Instanzen werden angezeigt. Zeigen die Pfeilspitzen voneinander weg, wird das Fenster vergrößert und alle Instanzen des Objekts werden angezeigt.

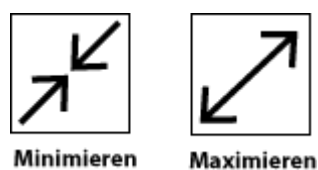


Abbildung 34: Mini-, Maximierenfunktion

Die Aufgaben eines Benutzers, aber auch die Objekte (bzw. Objektinstanzen) können sehr groß werden, weshalb ein Suchfeld viele Vorteile bietet. Der gesuchte Eintrag wird in das Eingabefeld eingegeben und mit Enter oder durch einen Klick auf das Lupen Symbol bestätigt.

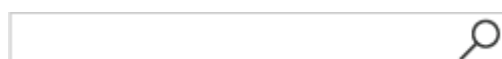


Abbildung 35: Suchfeld

7

Zusammenfassung und Ausblick

Im Rahmen dieser Arbeit wurde ein Usability-Konzept für die Laufzeitumgebung von PHILharmonicFlows entwickelt. Business Prozess Management Systeme existieren um die Arbeitsabläufe eines Unternehmens effektiv und effizient zu gestalten. Die Oberfläche von PHILharmonicFlows soll die Arbeitsabläufe eines Unternehmens unterstützen. Deshalb wurde darauf geachtet das die Oberfläche schnell, intuitiv und einfach zu bedienen ist. Es wurde mit einer flachen Hierarchie der Ebenen erreicht, dass man durch ein paar Klicks auf alle Inhalte der Oberfläche navigieren kann. Bei der Farbwahl des Designs wurde darauf geachtet, keine bunten Farben zu verwenden um der Oberfläche eine seriöses Auftreten zu geben. Die Oberfläche ermöglicht die Interaktion mit laufenden Prozessinstanzen über die erzeugten Formulare, Arbeitslisten und den Mikro-Prozess Graphen. Das entwickelte Usability-Konzept ist in sich geschlossen und stabil. Die Gestaltung des Makro-Prozesses wurde in dieser Arbeit außer Acht gelassen, da das Hauptmerkmal auf den Mikro-Prozessen lag. Der größte Teil der Oberfläche wurde bereits implementiert und kann als Screencast² aufgerufen werden [23].

2) aufrufbar auf <https://vimeo.com/130551330>

A

Mikro-Prozess Graph

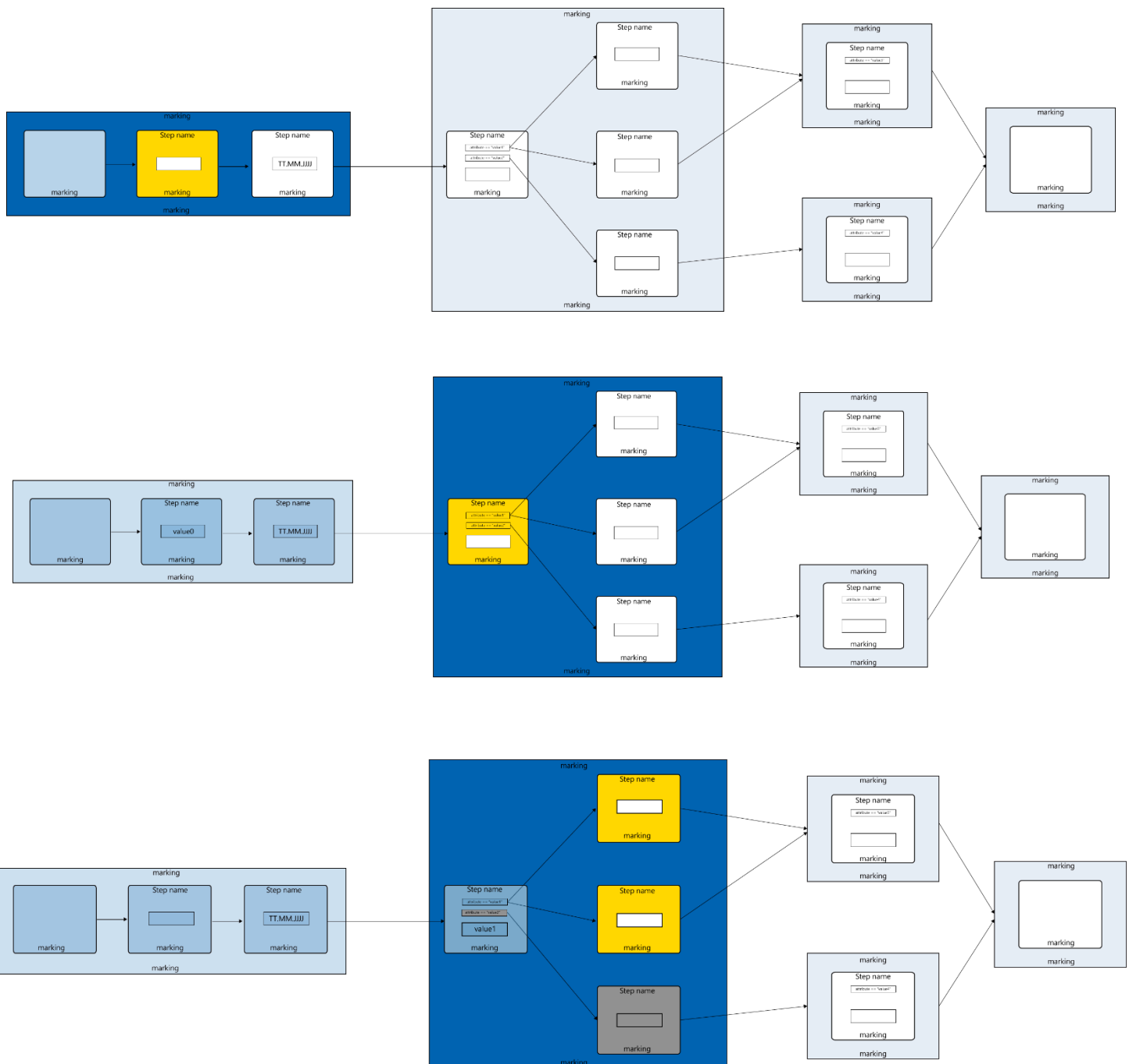


Abbildung 36: Mikro-Prozess Graph

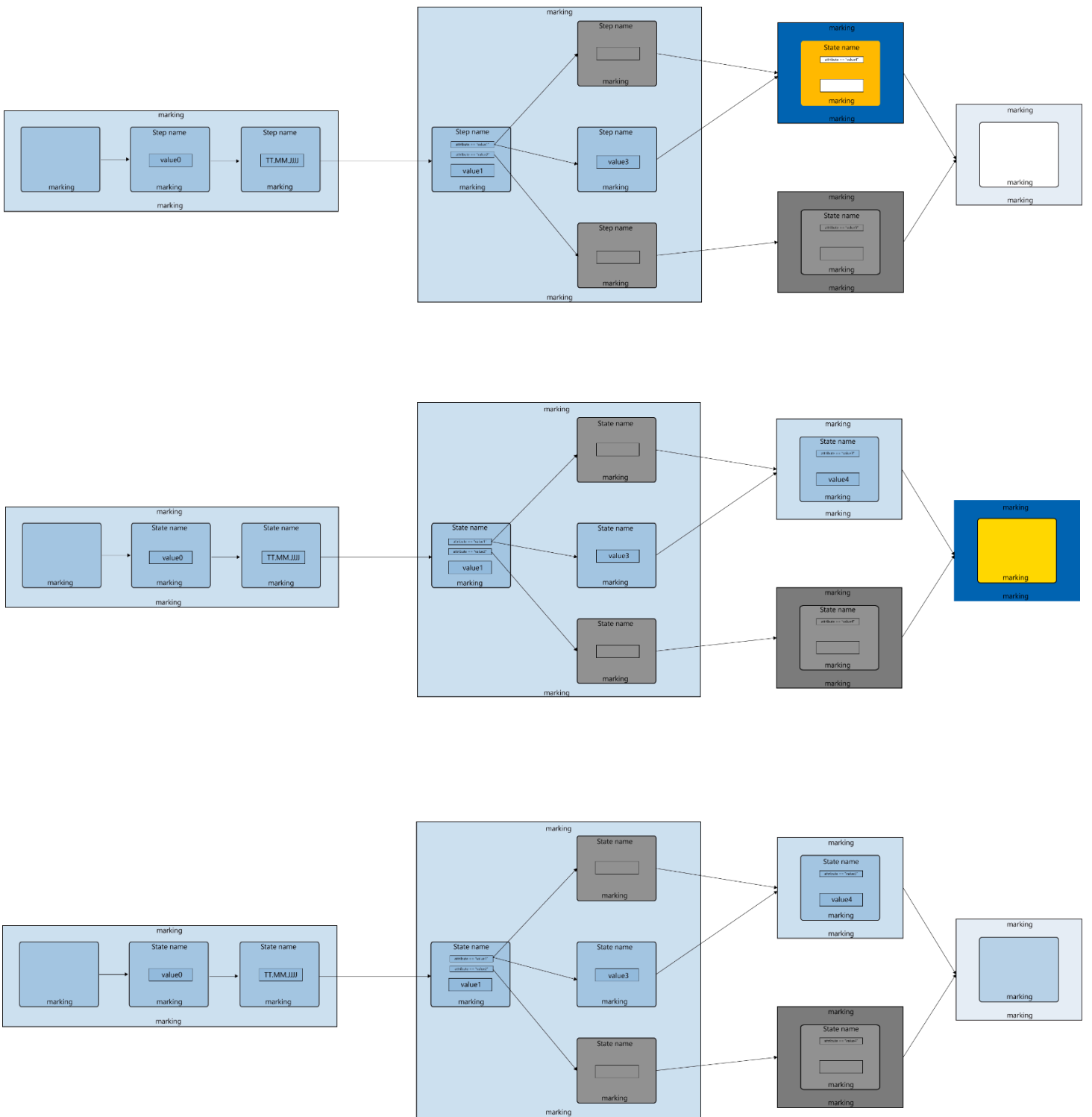


Abbildung 37: Fortsetzung Mikro-Prozess Graph

Abbildungsverzeichnis

Abbildung 1: Anforderungsanalyse	12
Abbildung 2: User-Interface-Entwurf.....	18
Abbildung 3: Die Hauptkomponente von PHILharmonicFlows	23
Abbildung 4: Beispiel für Datenstruktur	25
Abbildung 5: Beispiel Mikro-Prozess [20]	26
Abbildung 6: Generelle Bildschirmaufteilung	31
Abbildung 7: Sidebar Buttons (normaler Endbenutzer)	32
Abbildung 8: Sidebar Buttons (Administrator).....	32
Abbildung 9: Anmeldung PHILharmonicFlows.....	33
Abbildung 10: Status Informationen.....	34
Abbildung 11: Startbildschirm Administrator Ansicht	34
Abbildung 12: normaler Endbenutzer Ansicht.....	35
Abbildung: 13 Ebenen und Beziehungen	36
Abbildung 14: Objekte Verknüpfen	37
Abbildung 15: Ansicht Objektverwaltung	37
Abbildung 16: Ansichten Mikro-Prozess Darstellung	38
Abbildung 17: Ansicht Mikro-Prozess Darstellung	39
Abbildung 18: Prozess zur Bearbeitung ausgewählt	39
Abbildung 19: Baumstruktur	40
Abbildung 20: Button Minimieren und Maximieren	40
Abbildung 21: Komplexeres Beispiel eines Objektbaumes mit minimiertem Objekt	41
Abbildung 22: Farben Schritte Mikro-Prozess	43
Abbildung 23: Ausschnitt Mikro-Prozess	43
Abbildung 24 Hinweisdarstellung im Formular und Graphen.....	44
Abbildung 25: Farbwahl Startseite.....	46
Abbildung 26: Farbwahl Anmeldung.....	46
Abbildung 27: Objektverwaltung	48
Abbildung 28: Verknüpfte Objektansicht.....	48
Abbildung 29: Schriftgröße und -schnitt.....	49
Abbildung 30: Ausschnitt Sidebar.....	50
Abbildung 31: Icons die Aktionen auslösen	50
Abbildung 32: Ansichten Mikro-Prozess Darstellung	51

Abbildung 33: Statusinformationen.....	51
Abbildung 34: Mini-, Maximierenfunktion.....	51
Abbildung 35: Suchfeld.....	51
Abbildung 36: Mikro-Prozess Graph.....	53
Abbildung 37: Fortsetzung Mikro-Prozess Graph.....	54

Tabellenverzeichnis

Tabelle 1: Grundsätze der Dialoggestaltung.....	6
Tabelle 2: Allgemeine Usability-Heuristiken.....	7
Tabelle 3: Designprinzipien von Microsoft	8
Tabelle 4: Farben und ihre Bedeutung	11
Tabelle 5: Laufzeitmarkierungen eines Zustandes	27
Tabelle 6: Laufzeitmarkierungen eines Schritts	28
Tabelle 7: Zustandsmarkierung	42
Tabelle 8: Farben Allgemein	47

Literaturverzeichnis

- [1] V. Künzle und M. Reichert, „Herausforderung bei der Integration von Benutzern in Datenorientierten Prozess-Management-Systemen - EMISA Forum,“ 2011.
- [2] Standard Coordinating Committee of the Computer Society of the IEEE, „IEEE Standard Glossary of Software Engineering Terminology,“ IEEE Standards Board, Usa, New York, Std. 610.12-1990.
- [3] P. Hartung, Arbeitssicherheit und Gesundheitsschutz im Büro, Band 1, Bd. 1, P. Hartung, Hrsg., WEKA Media GmbH & Co KG, 1997, 1997.
- [4] C. Rudlof, „Handbuch Software-Ergonomie Usability Engineering,“ 2006. [Online]. Available: <http://www.ukpt.de/pages/dateien/software-ergonomie.pdf>. [Zugriff am 31 August 2015].
- [5] M. Venzin, „somedia PRODUCTION PRINT VIDEO WEB,“ 23 September 2013. [Online]. Available: <http://blog.somedia-production.ch/online/wireframe-mockup-prototype-was-ist-der-unterschied>. [Zugriff am 8 September 2015].
- [6] M. Dahm, Grundlagen der Mensch-Computer-Interaktion, München: Pearson Studium, 2006.
- [7] P. F. Elzer, Management von Softwareprojekten: Eine Einführung für Studenten und Praktiker, Springer-Verlag 2013, 2013.
- [8] DAkkS Deutsche Akkreditierungsstelle, „Leitfaden Usability,“ 14 Juni 2010. [Online]. Available: http://www.dakks.de/sites/default/files/71_sd_2_007_leitfaden_usability_1.3_0.pdf. [Zugriff am 2 September 2015].

- [9] N N/g Nielsen Norman Group, „10 Usability Heuristics for User Interface Design,“ 1 Januar 1995. [Online]. Available: <http://www.nngroup.com/articles/ten-usability-heuristics/>. [Zugriff am 13 September 2015].
- [10] Microsoft, „Windows 8 Design and coding guidelines,“ 13 Mai 2014. [Online]. Available: <https://msdn.microsoft.com/de-de/library/windows/apps/hh465424.aspx>.
- [11] R. Bart, „Visuelle Wahrnehmung,“ 3 März 2009. [Online]. Available: <http://www.robaweb.de/gdm/inhalt/VisuelleWahrnehmung/01-Einfuehrung.html>. [Zugriff am 13 September 2015].
- [12] C. Moser, User Experience Design, Berlin Heidelberg: Springer-Verlag, 2012.
- [13] I. Behle, Expertensystem zur formalen Werbebildgestaltung: Beratung zu Techniken und Werbewirkungspotentialen, Springer-Verlag, 2013, p. 135.
- [14] C. Graack, „KOMPAKTDESIGN Werbeagentur des Mittelstands,“ 5 August 2013. [Online]. Available: http://www.kompaktdesign.de/farben_in_der_werbung/. [Zugriff am 14 September 2015].
- [15] M. Offergeld, *Vorlesungsskript Usability Engineering*, Universität Ulm, 2013.
- [16] ergo-online, „ergo-online,“ 16 August 2007. [Online]. Available: <http://www.ergo-online.de/site.aspx?url=html/rechtsgrundlagen/bildschirmarbeitsverordnung/gestaltungsregeln.htm&qu=gestaltungsregeln>. [Zugriff am 11 November 2015].
- [17] Microsoft, „Was ist eine App der universellen Windows-Plattform (UWP)?,“ 2015. [Online]. Available: <https://msdn.microsoft.com/de-de/library/windows/apps/dn726767.aspx>. [Zugriff am 18 September 2015].

- [18] R. Stokes, „Emarketing: The Essential Guide to Digital Marketing,“ Porcupine Press, 2011.
- [19] A. Pröbstle, *Technische Konzeption und Realisierung der Modellierungskomponente für ein datenorientiertes Prozess-Management-System - Diplomarbeit*, Universität Ulm, 2011.
- [20] V. Künzle, „Object-Aware Process Management - Dissertation,“ Universität Ulm, April 2013.
- [21] S. Schultz, „Implementierung einer Komponente zur Ausführung von Mikro-Prozessen in einem datenorientierten Prozess-Management-System - Diplomarbeit,“ Universität Ulm, 31 März 2012.
- [22] H. Beck, „Implementierung einer Komponente zur Modellierung von Mikro-Prozessen in einem datenorientierten Prozess-Management-System - Diplomarbeit,“ Universität Ulm, 2012.
- [23] DBIS Ulm, „[Screencast]PHILharmonicFlows Runtime Demonstration Tool,“ 2015. [Online]. Available: <https://vimeo.com/130551330>.

Name: Angelina Henschel
776711

Matrikelnummer:

Erklärung

Ich erkläre, dass ich die Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

Ulm, den
Angelina Henschel